REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE Dec 94	3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE
Visualization of Finite Element Model Solutions for Defibrillation Voltage and Current Distributions

5. FUNDING NUMBERS

6. AUTHOR(S)

Steven Jonathan Jantz

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

AFIT Students Attending:

University of Washington

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/CI/CIA

94-161

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

DEPARTNEMT OF THE AIR FORCE
AFIT/CI
2950 P STREET, BDLG 125
WRIGHT-PATTERSON AFB OH 45433-7765

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

DTIC
ELECTE
JAN 18 1995
S G D

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for Public Release IAW AFR 190-1
Distribution Unlimited
BRIAN D. GAUTHIER, MSgt, USAF
Chief Administration

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

19950117 011

DTIC QUALITY INSPECTED 3

14. SUBJECT TERMS			15. NUMBER OF PAGES 62
			16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

DISCLAIMER NOTICE
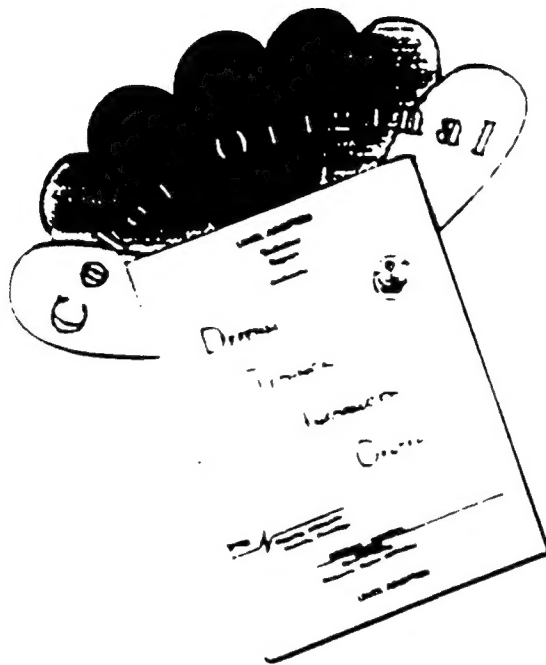
THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF COLOR PAGES WHICH DO NOT REPRODUCE LEGIBLY ON BLACK AND WHITE MICROFICHE.

Visualization of Finite Element Model Solutions
for Defibrillation Voltage and Current Distributions

by

Steven Jonathan Jantz

A thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science in Electrical Engineering

University of Washington

1994

Approved by _____
Chairperson of Supervisory Committee

Program Authorized
to Offer Degree _____ Electrical Engineering_____

Date _____

Master's Thesis

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this thesis is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Any other reproduction for any purposes or by any means shall not be allowed without my written permission.

Signature _____

Date _____

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

Sudden cardiac death, due to ventricular arrhythmias, claims the lives of 350,000 to 400,000 people in the United States each year [1]. For those that survive, and others at high risk of cardiac arrest, efforts are made to protect against sudden cardiac death in the future. Implantation of transvenous cardioverter-defibrillators is rapidly becoming a "first-choice approach" in the treatment of patients with these life-threatening arrhythmias [2]. Unfortunately, the implantation procedure involves repeated induction of ventricular fibrillation, and subsequent defibrillation, in order to determine a suitable electrode configuration and shock energy for each patient [3]. These tests greatly increase the trauma and risk to the patient. It would be very advantageous to have some patient-specific information, prior to the surgery, to aid in the placement of the electrodes.

The finite-element method (FEM) is a numerical analysis tool that can be used to calculate the voltage and current distributions in the thorax that are produced by a defibrillation shock [4]. We have demonstrated a high correlation between the voltage predictions of a high-resolution FEM model and experimentally-measured voltages in pigs [5]. We are currently working to refine, and improve our modeling process in order to increase its usefulness. The model is constructed from a series of classified computed tomography (CT) images of an individual. The classification is done by a semi-automated image classification program which identifies each tissue type in the thorax, including the various electrodes [6]. These tissue types are given appropriate conductivity values, and a 3-dimensional model is constructed composed of many small, linear, hexahedral elements of identical size and shape. Electrode boundary conditions are set, and the FEM is used to solve for the voltages throughout the thorax. From these voltages and the tissue conductivities, the voltage gradients and current densities are also computed. The results of these computations can then be studied in various ways. For example, once the model has been solved, the user can define a region of interest, such as a coordinate cube, or all elements of a given tissue type, and then output the voltage, voltage gradient, or current density data from that region to a file. Using a spreadsheet or other statistical tools, this

data can be further analyzed by, for instance, calculating the histogram of voltages in a given volume, or finding the average voltage gradient in a certain tissue.

Although much can be, and has been, gained from such analyses, our research can be improved significantly with the ability to graphically visualize the model and its results. This visualization capability will be useful in many different areas related to defibrillation research. We have identified four major areas. First, it will aid in the development and evaluation of new modeling techniques and solution algorithms. We are currently working on an adaptive meshing strategy to lead to a more efficient model solution [7]. Visualization can be used to compare the electric field solutions of this new technique to those of our existing model. Secondly, visualization will help in evaluating new electrode sizes, shapes, and configurations. We will be able to visually analyze the voltage and current distributions of these new electrodes and compare them to the distributions produced by electrode configurations now in use. A third area is in researching the variables leading to successful defibrillation. It is currently accepted that the magnitude of the voltage gradient in the myocardium is an effective indicator of the efficacy of a defibrillation shock [8]. This analysis is feasible with our current model solution program environment. However, it may be possible that the voltage gradient magnitudes, or even the gradient direction, in a certain area of the myocardium are more influential in affecting defibrillation than those in other areas. A visual analysis will be helpful in investigating this possibility, as well as in determining any other parameters which might be involved in indicating defibrillation efficacy. Finally, visualization will be useful in a clinical setting. We are working toward the generation of patient-specific FEM models for use in the defibrillator implantation process. Visualization will play a key role in determining the optimal electrode configuration and shock energy for each patient, as well as in aiding the doctor in the actual placement of the electrodes.

The objective of this project was to begin filling this need for visualization. A set of tools has been developed to aid in analyzing the results of our FEM computations. Tools facilitating the display of both scalar and vector quantities were included. The toolset was developed utilizing the graphics routines and features available in a

commercial scientific visualization package - the Application Visualization System (AVS) produced by Advanced Visual Systems, Inc. (Waltham, MA). The attempt was made to give the user as much control over the visualization environment as possible with a minimum amount of knowledge of the underlying software. The project was begun on an IBM RS6000 computer. During the toolset development, the work was ported to a SUN SPARCstation 20 and it was completed there.

This thesis describes the development of these visualization tools for use in the AVS environment. Chapter 2 discusses the previous efforts in this type of visualization, and introduces a set of visualization requirements that we set out to fulfill. Chapter 3 describes each of the tools available in the FEM visualization suite, along with the rationale for including each tool. A basic performance study is also presented including the time required for various calculations to be made in each of the visualization tools on both the IBM and SUN systems. Chapter 4 provides an evaluation, both objective and subjective, of the presented toolset. Chapter 5 concludes the thesis with some suggestions for future efforts in the improvement of these visualization tools.

Chapter 2: Visualization and Our Requirements

The discipline of scientific visualization has shown substantial growth over the last few years, especially as computer hardware configurations have been developed with increased memory capacity, fast CPU speeds, special purpose processors, and graphics accelerators. Ongoing and emerging research topics continue to build and shape it in incredible ways [9]. The use of parallel computing [10], as well as the use of virtual reality in scientific visualization [11] are examples of research areas which, when fully developed, may prove beneficial to our visualization application. However, due to constraints in both time and resources, we decided not to pursue these new research areas in this project. Based on the increasing capabilities of scientific visualization packages currently available, we chose to develop our visualization environment using one of these commercial systems. In this chapter, we discuss how other research groups visualize their models, and then present our approach with a set of visualization requirements.

2.1 Bioelectric FEM Visualization

Though there are currently significant efforts in researching the computer modeling of the bioelectric response in the thorax due to defibrillation shocks, very little information exists describing the methods used to visualize these computer models and their results. The different types of displays and visualization techniques utilized by researchers can be observed by reading through the literature presenting their progress, but the underlying visualization environment is rarely discussed. Many FEM models began as 2-dimensional (2-D) implementations. The visualization of these models is straightforward, since we are accustomed to visualizing in 2-D. Ramirez et al. [12] used color-coded "regional distribution" displays to show which areas of the myocardium (in the 2-D model) received a given range of current density values. Sepulveda et al. [13] used "isopotential contours" to show the voltage distribution in the slice, coupled with displays using "current lines" to trace through the 2-D current density vector field. These techniques have been carried over into 3-dimensional (3-D) modeling. For example,

Ideker et al. [8] included displays showing isopotential or isogradient lines drawn on the surface of the heart, while Tang et al. [14] decided to color the heart surface based on the voltage gradient field. However, even though they are using 3-D models, neither of these research groups attempted to make the heart look like a 3-D object in their displays. On the other hand, Karlon et al. [15] and Johnson et al. [16] used shading and/or perspective views to give their displays a 3-D look, and then used colors to map the distribution of voltage or current onto various anatomical surfaces. Still other researchers have investigated the mapping of cardiac activation sequences, i.e., the measurement of the time of electrical activation at multiple sites on the epicardial or endocardial surfaces of the heart. These mappings have usually been displayed as simple 2-D views with no depth cues [8,17,18]. However, some effort has also gone into presenting these mappings in 3-D projections or as volume rendered displays [17,18,19].

Only two instances were found in which an entire publication was devoted to the description of the visualization environment used in viewing 3-D bioelectric models and their results. The first environment, developed at the University of Utah, has been presented in a number of different articles [20,21,22]. The environment includes two sets of custom visualization tools based on Silicon Graphics' GL graphics library. The first set is a suite of interactive programs, known as "Map3d," used for quick display and editing of geometry and surface-rendered data. The second set of programs is based on ray-traced rendering and distributed computing and is used to generate presentation-quality images. The other documented visualization environment was developed by a research group at Duke University for use in visualizing 3-D cardiac activation mapping models. The environment, called "Vox," was designed to superimpose multivariate experimental data on complex bioelectric structures entirely by volume-rendering techniques [18]. A C-like language-based interface, Vexpr, was developed to allow the investigators to specify mappings (for shading, opacity, and color) from the multivariate volume data to rendered images. The output from Vexpr is sent to a hybrid ray tracer, Vtrace, that can render both volumetric and polygonal representations.

The researchers developing each of these environments did much of their own programming, designing the tools solely for the purpose of displaying these bioelectric computer models. We decided not to take this approach in developing our visualization tools, but, instead, made use of an existing commercial scientific visualization package. This package allowed for the expeditious development of our visualization environment and included features that helped us satisfy our visualization requirements.

2.2 General Visualization Requirements

We began by defining various high-level requirements for our FEM visualization environment. The first requirement is ease of use. Everyone from students to radiologists and cardiologists will want to use visualization at some point, so the system should be accessible to all. Another requirement is flexibility. As our visualization needs change and expand, the software environment should make those necessary changes and upgrades not only possible, but easy. Thirdly, we require a portable system. Since we do not know the platform that the clinical visualization programs will ultimately be run on, the system will need to be easily ported to a variety of platforms. These same requirements were some of the goals incorporated into the development of the Application Visualization System (AVS), produced by Stellar Computer (now Advanced Visual Systems, Inc.) [23]. AVS is a modular system which allows software components (modules) to be combined into an executable flow network. Each module incorporates user-controlled parameters in the implementation of a specific visualization function, e.g., data input, filtering, mapping, or rendering. Users can then design application-specific networks that meet their visualization needs.

2.3 Specific Visualization Requirements

Once we had chosen the AVS package as a base for our visualization environment, we defined a set of specific requirements to guide our toolset development. These requirements are divided into two categories. The first is visualization and manipulation of electrodes in the classified tissue model before it is solved. The other is visualization and analysis of the voltage and current distributions after the model is solved.

2.3.1 Pre-solution visualization

For our research to be useful in investigating the expected efficacy of various electrode configurations, it must be possible for us to insert electrodes into arbitrary locations of the classified model. We will require the ability to generate the geometric models for the various electrodes to be simulated, and then to integrate them into the classified slices of the thorax model. Using a combination of 2-D and 3-D image views, the user should be able to place, move, and deform the electrodes as necessary. The mouse and keyboard controlled movements must be updated simultaneously in all the views. Once the electrodes are in the desired location, the data should be output as new classified tissue slices to be used in the FEM calculations. This pre-solution visualization idea is still in its concept stages and was not dealt with in this project.

2.3.2 Post-solution visualization

Our immediate visualization needs focus on the desire to view the results of the FEM computations. In light of the areas of FEM/defibrillation research in which visualization will be helpful (see Chapter 1), we identified various requirements for our suite of visualization tools:

1) Given the data file containing the results of the FEM calculations, we must be able to generate 3-D data sets suitable for AVS. The required data sets include the vector quantities of current density and voltage gradient, as well as the scalar quantities of the classified tissue types, voltages, and magnitudes of the vector quantities.

2) We need to be able to view the data in both 2-D and 3-D. 3-D visualization can provide an overall feel for the voltage and current distributions with respect to the various tissues and electrode locations in the thorax. The 2-D ability should allow further investigation by, for example, providing a way to view an arbitrary slice extracted from the model. It should be possible to combine these techniques, e.g., display a slice inside a 3-D view in order to show its orientation and position in the model.

3) We should be able to look at multiple images simultaneously. This will allow for comparisons of the data from the same or different FEM models.

4) We want to be able to extract a subset of the tissue types and display the surface defining those tissues. The color of the surface should be determined by the value of one of the scalar quantities at each point of the surface.

5) In addition to this tissue type extraction, we also should be able to define and display other Regions of Interest (ROIs), based either on the voltage or current values, or on a mouse-selected area.

6) We should have the use of a probe to allow us to find the data value at a specific location in the display. This capability should be in both the 2-D and 3-D tools. In multiple displays of the same data, we want to have concurrent movement of the probe in all the views.

7) In a view of a given slice of scalar data, we should be able to overlay an outline of all the tissue boundaries so we can see which tissues correspond to which values.

8) We need the capability of displaying both the magnitude and direction of our vector quantities in intuitive ways.

9) We should be able to incorporate numerical and statistical data with our visual data, e.g., show a histogram of the voltages in a given ROI.

10) We should be able to control various display features, such as the color range, the transparency of an object, and the viewing angle.

11) We want to be able to generate a color hardcopy output of the images for archiving and publications.

12) We need an intuitive and easy, preferably menu-driven, user interface to the visualization tools.

13) We want to be able to use the tools interactively. If the inconvenience of spending time waiting for new displays to be calculated outweighs the benefits gained from the visualization, then the environment will not be very useful. Based on our experience manipulating models in AVS on a SUN SPARCstation 20, two standards were set that quantified "interactive" for our application. First, when an object is rotated, the display of each new viewing angle should take less than 15 seconds to update. Second, after changing a parameter or choosing a different data field to view, the new

calculations should be made and the display updated in less than 30 seconds. In order to further qualify these times and allow for comparison to other machines, benchmark results for the SUN are presented in the performance discussion in Section 3.5.

With these requirements in mind, we set out to assemble an application visualization environment for our FEM model. This suite of tools is described in the next chapter.

Chapter 3: Implementation

In developing a suite of visualization tools, many different ways of displaying both vector and scalar quantities were investigated. The set of modules provided by AVS, along with those written by other AVS users (made available via anonymous ftp to the International AVS Center (IAC) at avs.ncsc.org), included most of the functions needed in order to construct our visualization suite. Given these modules, we had to first determine which of them were best suited for our viewing needs, then build the networks combining those modules appropriately, and finally design the presentation of the displays and controls for ease of use. In addition, some changes were made to existing modules, and new procedures and modules were generated, which were necessary in implementing or improving each visualization method. The tools that we decided to include were those which provided an analytically useful visualization method, presented a visually intuitive display of the model, and contributed to fulfilling our requirements. Before presenting this set of visualization methods, a description of the necessary conversions performed on our FEM data sets is included.

3.1 Data Set Conversion

Before data sets can be visualized in AVS, they must be converted into one of the formats required by AVS. There are two different AVS data types which are suitable for our FEM data. The first is known as "unstructured cell data" (UCD), which is best described as a coordinate structure of cells. The cells, which are defined by their corresponding nodes in the coordinate system, can be points, lines, quadrilaterals, triangles, tetrahedrons, pyramids, prisms, or hexahedrons. Data, either scalar or vector, can be associated with the entire structure, with each cell, or with each node. The other data type is called "field" data, which can be described as an n-dimensional array with an m-dimensional vector of values at each array location. In our case, we have a 3-dimensional array ($n=3$), with either a 1- or 3-dimensional vector at each location, depending on whether the field contains scalar quantities ($m=1$) or vector quantities

(m=3). Some previous work had been done in converting our data into field format. These field conversion programs were refined, and a new program was written to convert our data into UCD format. The use of nodes and cells in the UCD format makes it well suited for the element structure of FEM models. Since multiple data values can be assigned to each cell, UCD models are very flexible, especially in producing displays incorporating different data quantities, e.g., a surface based on tissue type but colored according to the voltage values. On the other hand, the cellular structure of the UCD data type also leads to a lengthy file format. It involves defining the coordinate values for each node in the model, followed by a list of the nodes that make up each cell, and finally defining all of the values assigned to each node and cell. Due to the size of our models (up to 2 million nodes), the UCD file size became unmanageable. However, since all of our elements are identical hexahedrons, they also fit nicely into the grid structure of the field data type. The data value for each element in our model is assigned to a point in the coordinate grid representing the center of that element. The field file format simply involves an ordered list of the data values at each point, cycling through each dimension in the field. After comparing the two data types, we decided to use multiple "field" files, one for each of the data quantities with which we are concerned. Thus, from one of the FEM data sets, four field files are generated: tissue, voltage, voltage gradient, and current density. Information on running the conversion program is contained in Appendix B.

3.2   User Interface

Numerous AVS networks have been designed to help in visualizing our data. We attempted to make the user interface for this visualization as friendly as possible. Once the FEM visualization application is started, the environment is menu driven. Thus, choosing a visualization method involves simple "point, click, and drag" mouse operations (Section A.7 describes the menu bar in detail). The menu remains on the screen with the chosen data viewing windows to allow for control over starting and stopping the flow of data, for returning to the top level menu where other viewing methods can be selected, or for quitting AVS. For each viewing method, the necessary parameter

controls are presented in the same window as the display, so that moving windows around on the screen is not necessary. The various controls used to adjust the parameters are intuitive and require simple mouse or keyboard operations (see Section A.5). Each of the visualization methods include a "data and color" window (Section A.4). This enables the user to choose which data fields they wish to view, and change that choice if desired. It also provides full control over the colormap used to assign colors to the values in the data sets. A color legend is displayed in each viewing method showing the color associated with each numerical data value in the model, as defined by the colormap. The color range is scaled to fit the range of values in the selected scalar field. When the vector tools are used, the colors are scaled to the range of magnitudes. Another helpful tool that we designed and included in each of the viewing methods is a small window that reminds the user of the basic mouse and keyboard controls for manipulation of the objects in the viewing windows. When further assistance is required in learning or using the tools, help can be found in the User's Guide, presented in Appendix A. The various user interface tools allow for easy control over the visualization methods. In the following sections we describe the different visualization methods that we developed based on the graphics capabilities available in AVS.

3.3 Scalar Visualization Tools

There are four scalar quantities in the FEM model that we wish to view. The first quantity is the classified tissue type as generated by the CT classification program. The tissue type is considered a scalar in AVS since an integer number is used to define each type. Using an intuitive color scheme, each tissue type is assigned a unique color to help in viewing the tissue field. The other three scalar quantities that we wish to view are results of the FEM calculations: the voltage, the magnitude of the voltage gradient vector, and the magnitude of the current density vector. A simple conversion network is included that calculates the magnitude of a vector field and writes the resulting scalar field as a new file. Since the scalar values used in the tissue fields are simply labels assigned to the various types, and since, for visualization, the tissue type data is used only as a reference

to the anatomical locations in the model, this data is in a separate class than the other three values. Thus, when "scalar" quantities or fields are referred to in the following visualization tool descriptions, this will include only the voltages and vector magnitudes.

3.3.1 3-slice 2-D

In the first viewing method, three slice planes are selected, one from each of the three cardinal dimensions of the model. These slices are extracted from the desired scalar field and are displayed as 2-D images. Each 2-D view has a probe connected to it so that the specific value at any location on the slice can be determined. Another viewer contains a 3-D projection of the rectangular bounds of the model, in which the slice planes extracted from the tissue type field are displayed. Thus, the user can simultaneously view the scalar values on the chosen slice planes, the tissue types in those slices, and the locations of the planes in the model (Figure 3.1). This provides a quick way to cycle
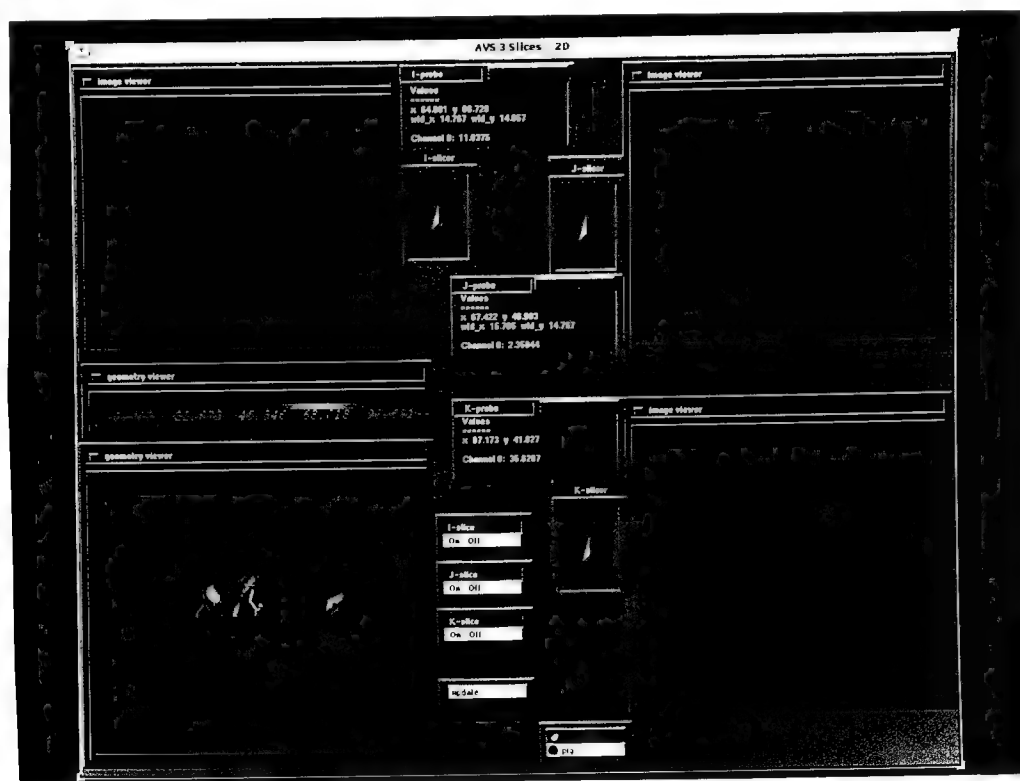

Figure 3.1: 3-slice 2-D display - three orthogonal slices
from the scalar field are shown in the 2-D images. The
tissue slices are presented in position in the 3-D model.

through all the slices in the model in order to get a feel for the voltage or current distributions in different areas of the thorax. If the 3-D view (which requires more time to compute than the 2-D view) is disconnected from the slice selection dials so that only the 2-D views are changed, these displays are immediately updated when a new slice is selected. Then, to refer back to the 3-D view, the tissue slices can be updated to the currently selected slices. This disconnection capability was added as a new module, since it is a useful viewing tool that was not originally available in AVS. Another ability that helps in analyzing the data is to be able to view one tissue slice at a time, so that the others do not obstruct the view. To allow for this, another module was written to allow the user to control the visibility of objects in the views, which, in this case, are the tissue slices. This visibility control module is also used in many of the other viewing techniques in different ways.

3.3.2   3-slice 3-D

This viewing method shows the same type of information as the last one, but it is presented in a different way. Since different users may have different preferences as to which method is more useful, both methods are included in the visualization toolset. As before, three slices are taken from the model. This time, however, the slices from the chosen scalar field are displayed in position in the 3-D projection. The corresponding tissue slices are then presented in the three 2-D views (Figure 3.2). This may be a more intuitive visualization method for some, since it displays the scalar values along with their position in the same viewing window. In this network, a probe is connected to the 3-D view since that is where the values of interest are. The user again has control over the slice numbers and their visibility in the 3-D model. Since the region of the heart is of most interest in defibrillation studies, a point-rendered surface (a surface that is displayed as points rather than as a solid object so that it does not obstruct anything that may be "behind" it) of the heart is displayed in position in the 3-D view in order to help connect the scalar values to the anatomy. This surface can be turned on and off as desired.
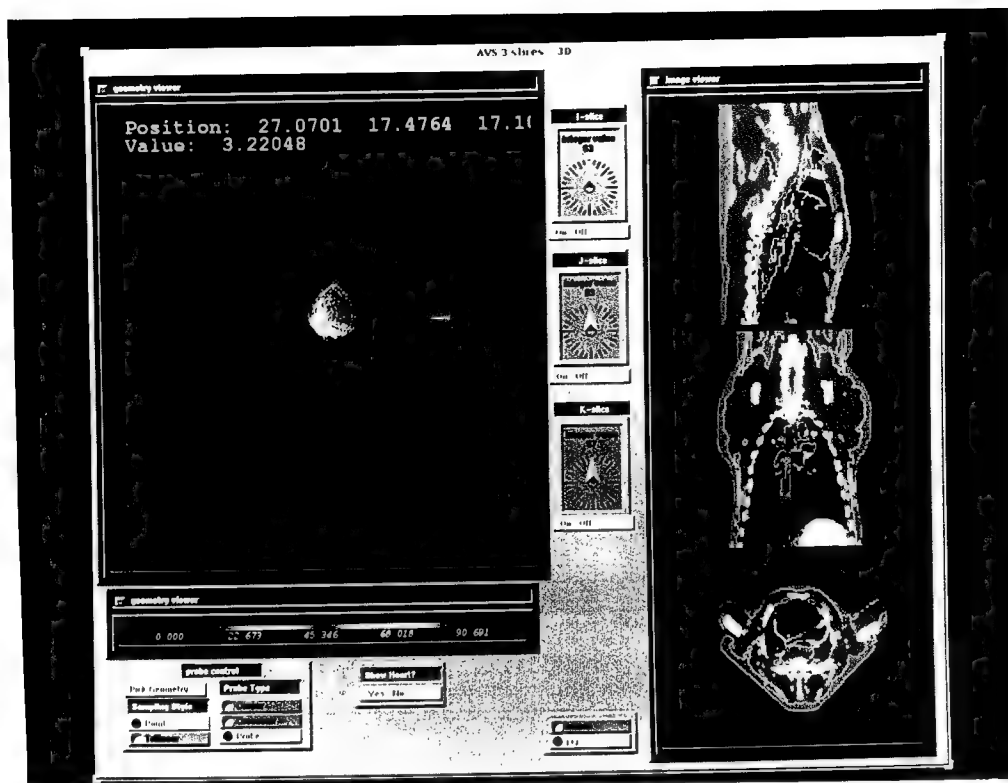
Figure 3.2: 3-slice 3-D display - three orthogonal slices
from the scalar field are shown in position in the 3-D
model. The tissue slices are presented as 2-D images.

3.3.3 3-slice 2-D with tissue outlines

This tool is very similar to the method in Section 3.3.1. It includes the same four

views. The 3-D view is identical, with control over the slice visibility. The only

difference is in the three 2-D views of the scalar field slices. In our defibrillation analysis,

we need to be able to determine the values of the various scalar quantities in specific

tissues, especially in the myocardium. Thus, in this viewing method each of the slices is

overlaid with an outline of the tissue boundaries for that slice (Figure 3.3). Used in

conjunction with the probe and the tissue slices displayed in the 3-D projection, the value

at a specific point on the slice can be identified to be in a specific tissue. The tissue

outlines are calculated by a simple contouring algorithm. Since the tissue types are

identified by consecutive integers, a contour line at each of these integer values will

effectively be the boundaries between the tissues. The user can choose the number of

contour lines drawn, as well as the starting value and the step size between lines. An extra
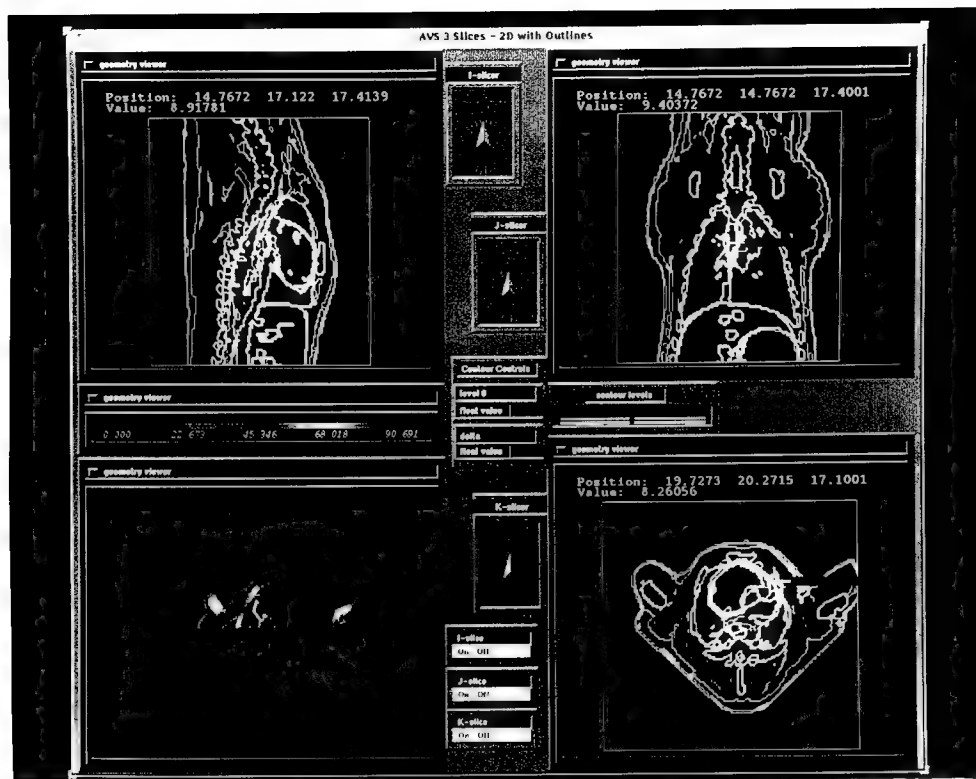
Figure 3.3: 3-slice 2-D with tissue outlines display - the 2-D slices from the scalar field are overlaid with an outline of the tissue boundaries. The tissue slices are in the 3-D display.

window is included listing the numbers associated with each tissue type to aid in this selection.

3.3.4 3-slice 3-D with tissue outlines

As the name suggests, this method adds the outline capability to the 3-slice 3-D network described in Section 3.3.2. Thus, the scalar field slices in the 3-D projection are overlaid with the tissue outlines (Figure 3.4). Again, this gives a more precise location to the values being viewed in the model. The probe remains available, as do the various controls on the tissue boundary calculations.

3.3.5 Arbitrary slicer

The first four tools, as well as others to come, require that each slice be parallel to one of the sides of the rectangular bounds of the model. This is easy to visualize, and easy
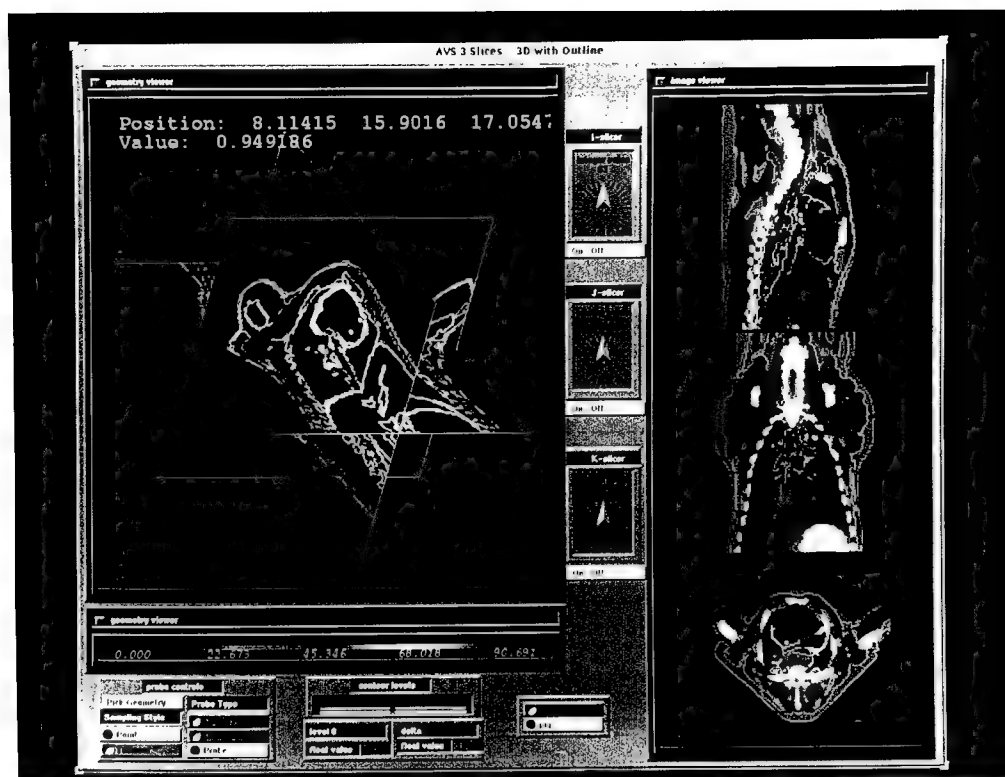
Figure 3.4: 3-slice 3-D with tissue outlines display - the
scalar field slices in the 3-D display are overlaid with
the outlines of the tissue boundaries. The tissue slices
are in the 2-D views.

for the software to compute. However, there may be times when we wish to view an arbitrary, oblique slice through the model. For example, we may wish to view the plane that slices the heart along its own major axis, which is not the same as any of the model's axes. This is the capability made available through this viewing tool. Two viewer windows are presented side by side (Figure 3.5). A point-rendered surface of the thorax is displayed to give an orientation of the model. The arbitrary slice plane through one of the scalar fields is then displayed in this same model view. In the other viewer, the same arbitrary slice extracted from the tissue field is displayed in the volume bounds. The slice plane location is controlled by defining its rotation and distance from the center of the model. Also, once the orientation of the model has been determined, the thorax surface can be turned off to give a totally unobstructed view of the chosen slice.
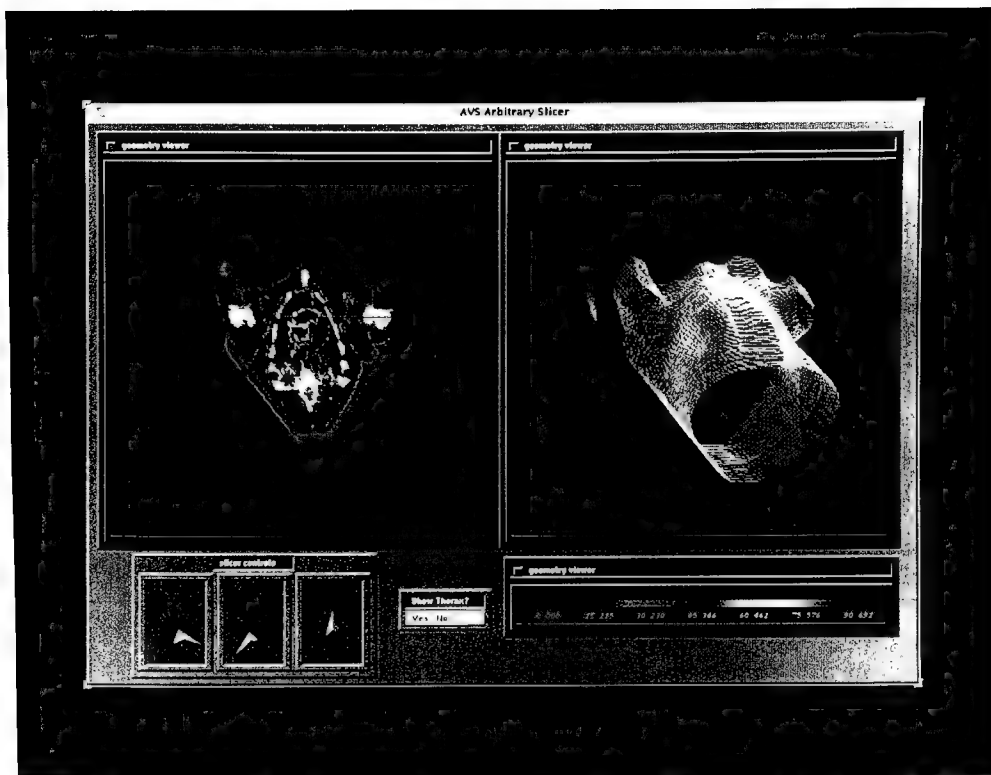
Figure 3.5: Arbitrary slicer display - An arbitrary, oblique
slice is extracted from both the scalar field and the tissue
field and displayed side by side. A point rendered tho-
rax surface is included in the scalar field display.

3.3.6 Colored tissue surface

In this visualization method, the tissue field is used to generate a surface
surrounding one or more of the defined tissue types. The user defines which tissues to
include in the surface by selecting the integer or range of integers that are assigned to the
desired tissues. Multiple ranges or nonconsecutive tissue numbers are not currently
supported. A threshold routine sets the value of any tissue that is not in the selected range
to zero. Then a 3-D extension of a 2-D contouring algorithm is performed which
generates a surface (known as an "isosurface") between all data points which have a zero
for their tissue type and those points which have a non-zero tissue type. The surface is
then colored according to the value of the selected scalar field at each point on the surface
(Figure 3.6). This type of display helps in investigating the flow of current into or out of a
given tissue. A point-rendered thorax surface is displayed for orientation, and can be
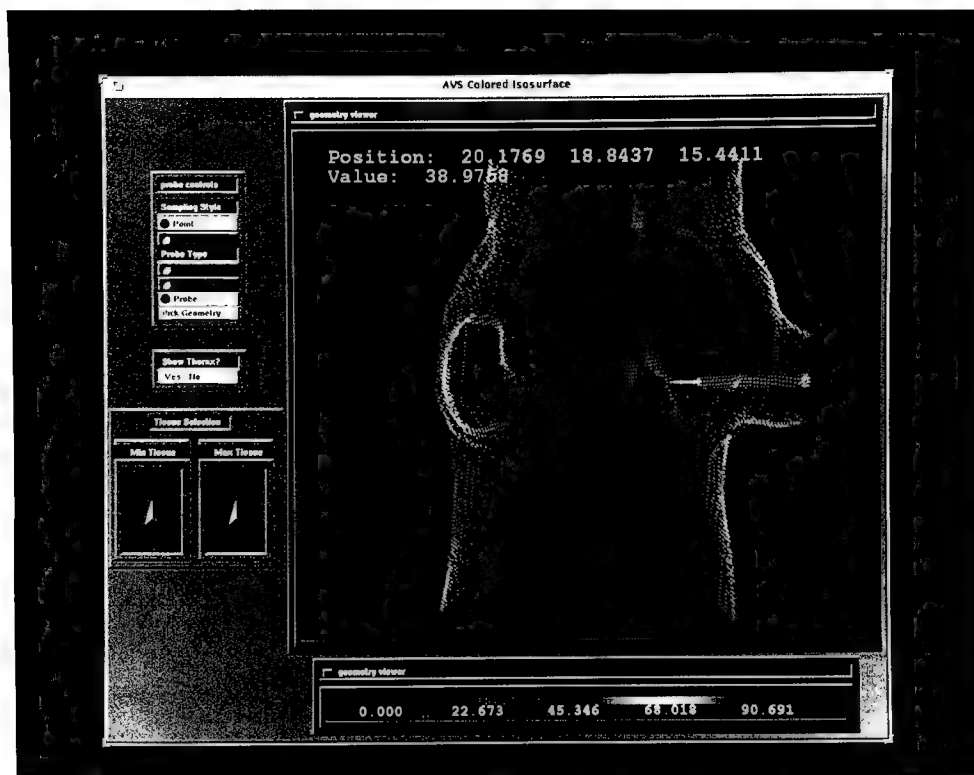
Figure 3.6: Colored tissue surface display - The surface sur-
rounding a selected range of tissues is displayed, col-
ored according to the scalar field values at each point on
the surface. A point rendered thorax surface is included.

turned on or off as desired. The tissue list window is provided to aid in choosing which
tissue(s) are to be included in the surface. A probe is available to allow for the
determination of values at specific points on the surface.

3.3.7 Heart cut-away

This tool arose from the desire to isolate the heart and perform a detailed analysis
of the voltage gradient and current density values in the myocardial tissues. It includes
four views in which only the surface of the heart is included (Figure 3.7). Each surface is
generated as described in Section 3.3.6 with the myocardial tissues automatically selected
for the surface. However, the user now has control over clipping planes which limit the
area of the model included in the display. Thus, the heart can be "sliced open" to reveal
the inside surfaces of the myocardial tissues. This viewing method is initialized such that
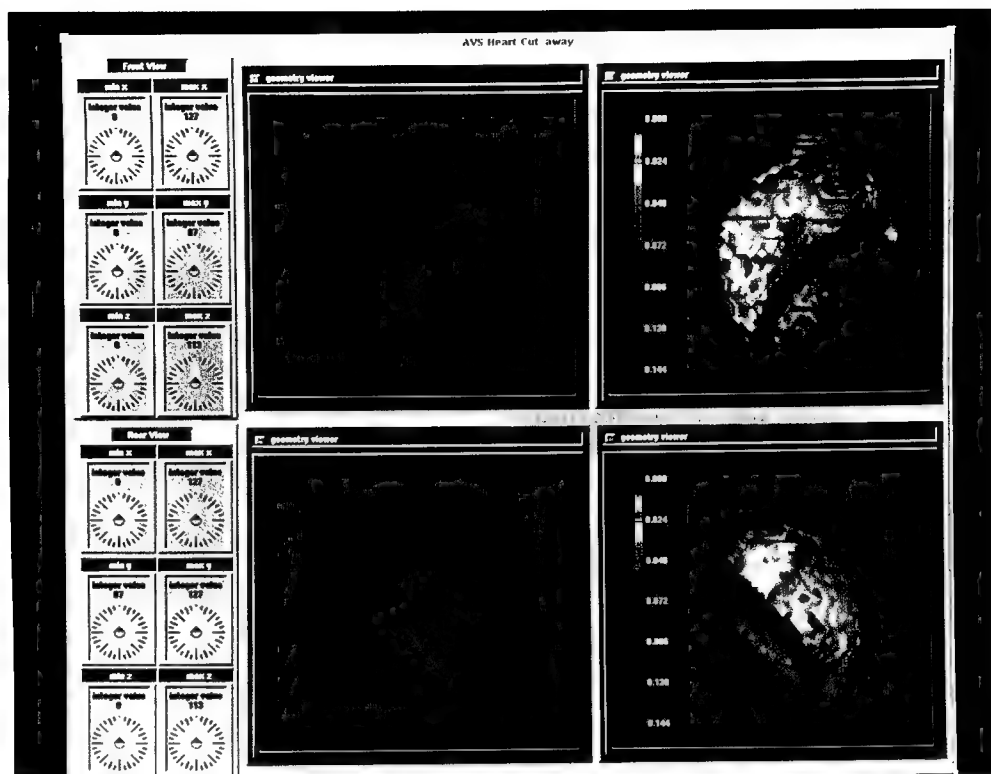
Figure 3.7: Heart cut-away display - the heart is sliced open
to reveal the inner surfaces of the myocardial tissue.
The top two views represent a front view and the bottom
two views show a rear view. The left views are colored
by tissue type: blue = atrium, green = right ventricle, red
= left ventricle. The right views are colored according
to the scalar quantities.

the top two displays represent views from the front of the thorax, and the bottom two are
views from the rear of the model. Thus, in the front views, the part of the heart nearest the
viewer (the anterior portion of the heart) is removed, and the inside surfaces of the back of
the heart are displayed. Likewise, in the rear views, the posterior portion of the heart is
removed, and the display shows the inside surfaces of the front of the heart. To help
identify what is being viewed, the left display of each of the front and rear views shows
the tissues colored according to their type, i.e., blue for atrium, red for left ventricle, and
green for right ventricle. The right display shows the same surface, but colored according
to the values of the chosen scalar field at each point on the surface. If desired, the outside

surface of the heart can also be viewed in these displays by simply rotating each surface 180 degrees.

3.3.8 Excavated thorax surface

This tool probably provides the most intuitive display of the voltage and current distributions, together with the area of the model that is being investigated. Only one viewing window is used, in which a solid surface of the thorax is displayed (Figure 3.8). Out of one corner of the thorax, a rectangular block is removed. The entire surface, including the interior planes resulting from the removed area, is colored according to the values of the scalar field. The user has control over the size of the excavated block, as well as which corner it is taken from. A point-rendered surface of the heart is included in the display which, again helps connect the values to the anatomy when viewing the model. A probe is available, as is control over the visibility of the heart surface.
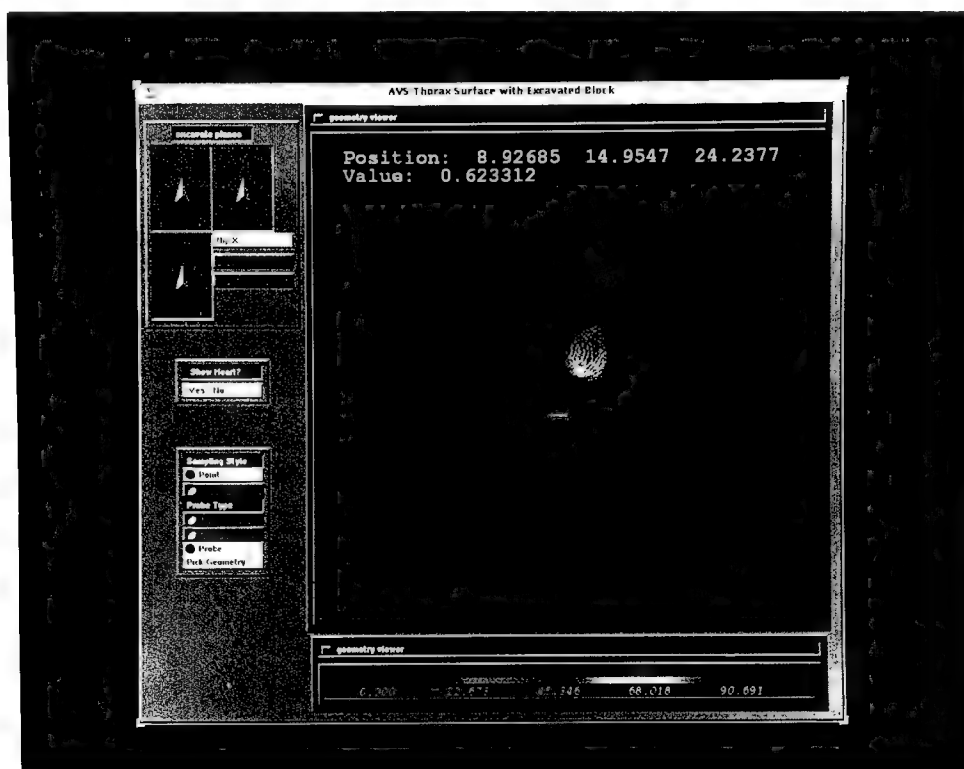
Figure 3.8: Excavated thorax surface display - the surface of the thorax is displayed with a rectangular block removed from one corner of the model. The entire surface is colored according to the scalar field.

3.3.9 Excavated volume bounds

This method is similar to the previous one. It removes a rectangular block from one corner of the model. Instead of displaying the surface of the thorax with the excavation, this method displays the surfaces defining the rectangular extents of the model minus the extracted block (Figure 3.9). Again, the heart surface, 3-D probe, and excavation controls are included in the network. Although this method is not as visually intuitive as the thorax display described in Section 3.3.8, this type of display proved to be much quicker than the tissue surface generation procedure - about twice as fast for rotations, and four times as fast for recalculations with new parameters (see Section 3.5).
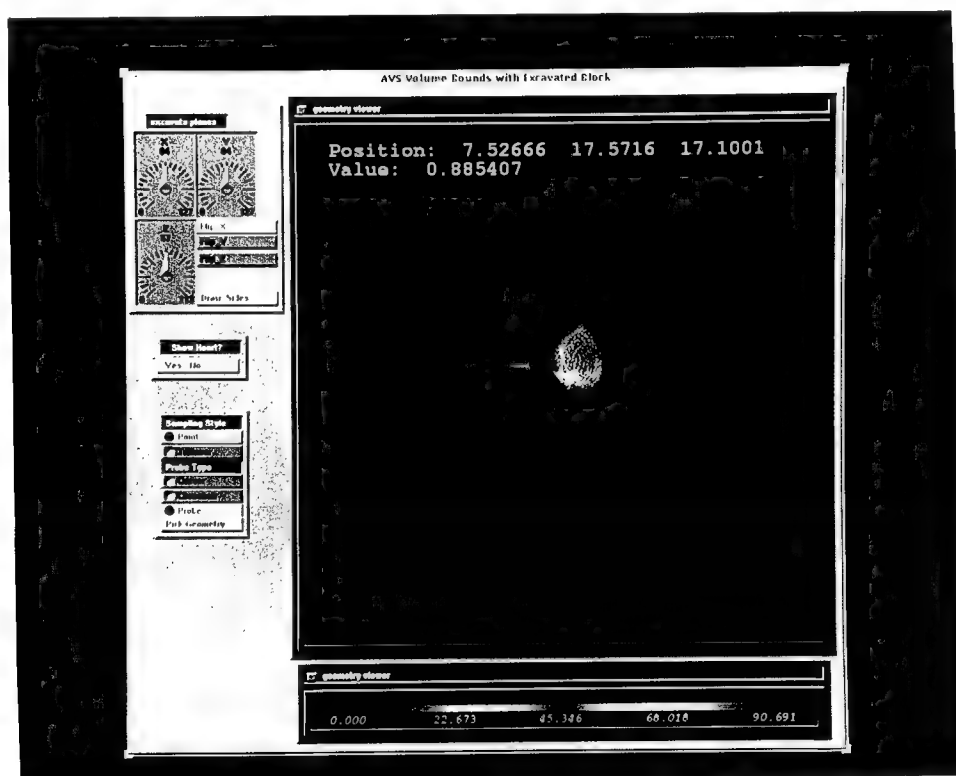
Figure 3.9: Excavated volume bounds display - the outer
planes of the model bounds are displayed with a rectan-
gular block removed from one corner of the model. The
entire surface is colored according to the scalar field.

3.3.10 Ray trace

This tenth visualization method is a volume rendering method known as ray tracing. The idea is that each volume element (voxel) in the model is given a color value and an opacity value as defined in the colormap. A ray is "shot" into the model from each pixel in the 2-D output image. Each voxel that the ray passes through makes a contribution to the color of the output pixel proportional to the voxel's opacity value. The ray travels through the model, accumulating color contributions, until the total opacity of all the voxels it has passed through is 1.0. The result is a 2-D image display which gives a 3-D feel for the voltage or current distribution being viewed (Figure 3.10). The field can be cropped (using clipping planes like those in Section 3.3.6) in any of the three dimensions, to allow viewing of only selected regions of the model. The field values can also be thresholded to limit the range of those values being included in the display. Other
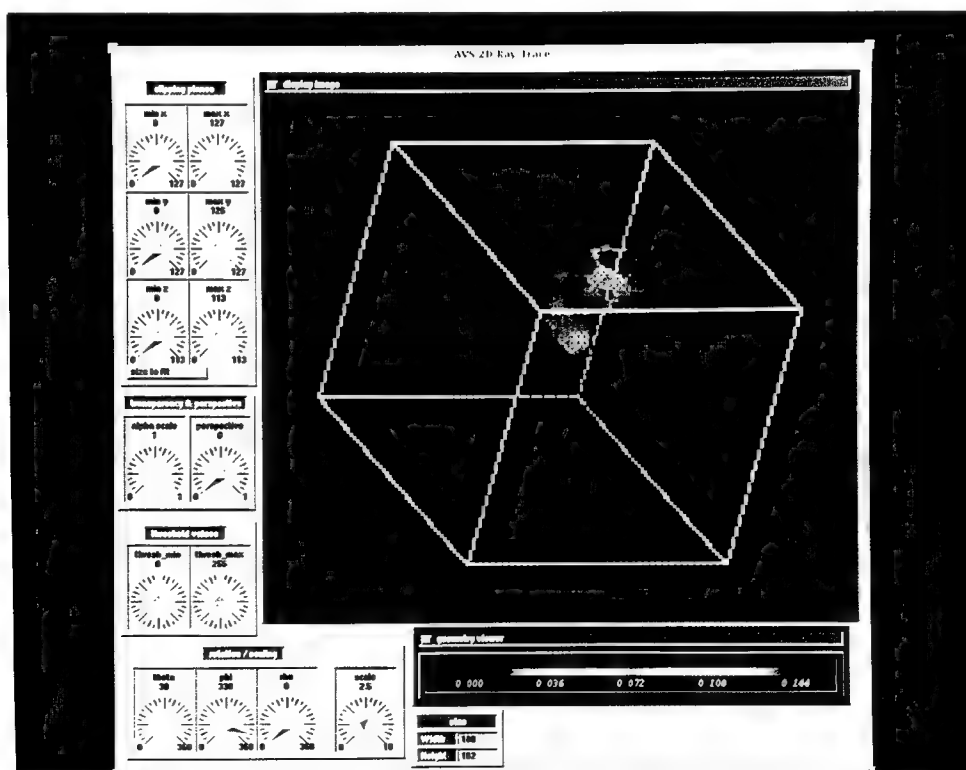
Figure 3.10: Ray trace display - a 2-D ray traced rendering of the scalar quantity field is displayed inside the model bounds.

controls allow the user to set the overall transparency of the display as well as define whether the display is shown as a perspective view; that is, displayed such that objects "close" to you are bigger than those "far" from you. The colormap can be used to further define the opacity of the colors used in the display. The size of the viewer can be changed as can the orientation of the model as it appears in the 2-D viewer.

3.3.11 Volume renderer

This is another volume rendering tool, similar to the ray trace. In this method, each voxel is assigned a color value and an opacity value based on the scalar field and a user-defined colormap. The volume is then displayed as a 3-D projection, such that the opacity values determine whether a voxel will block the voxels that are "behind" it in a given viewing angle. This method allows for incorporation of other 3-D visualization techniques into the display. The surface of the heart is included (Figure 3.11) as a
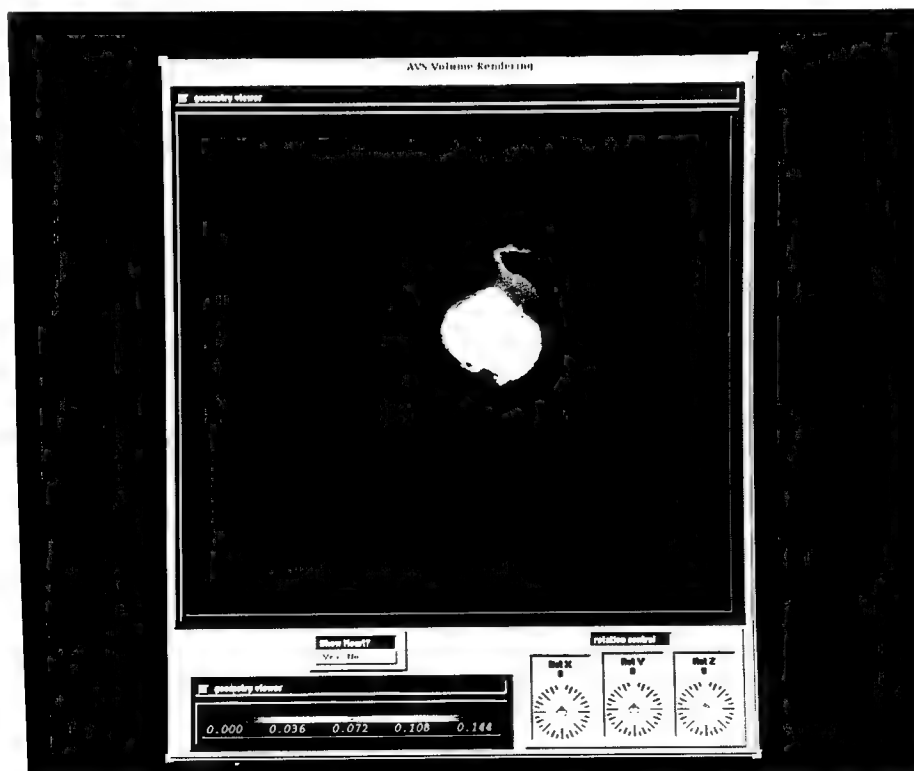
Figure 3.11: Volume render display - a 3-D volume rendered view of the scalar quantity field is displayed. A semitransparent heart surface is included.

reference for the orientation of the model and as an aid to determining the values of the scalar field in the heart region. The user has control over the visibility of the heart surface. This technique is very powerful in that it can combine surface rendering and volume rendering into one display, but it is also very compute-intensive, and calculating a new display for a rotation or new parameters takes over ten times as long as the ray tracing volume rendering tool (see Section 3.5).

3.3.12 Compare fields

The final scalar viewing tool is a very simple one. It allows for the comparison of scalar fields, which may come from the same or different models. Up to three fields can be selected and viewed simultaneously. A single transverse slice is taken from each field, and displayed in its own 2-D image viewer (Figure 3.12). Three color legends are shown, each scaled to the range of values in its corresponding field, and displayed beneath the
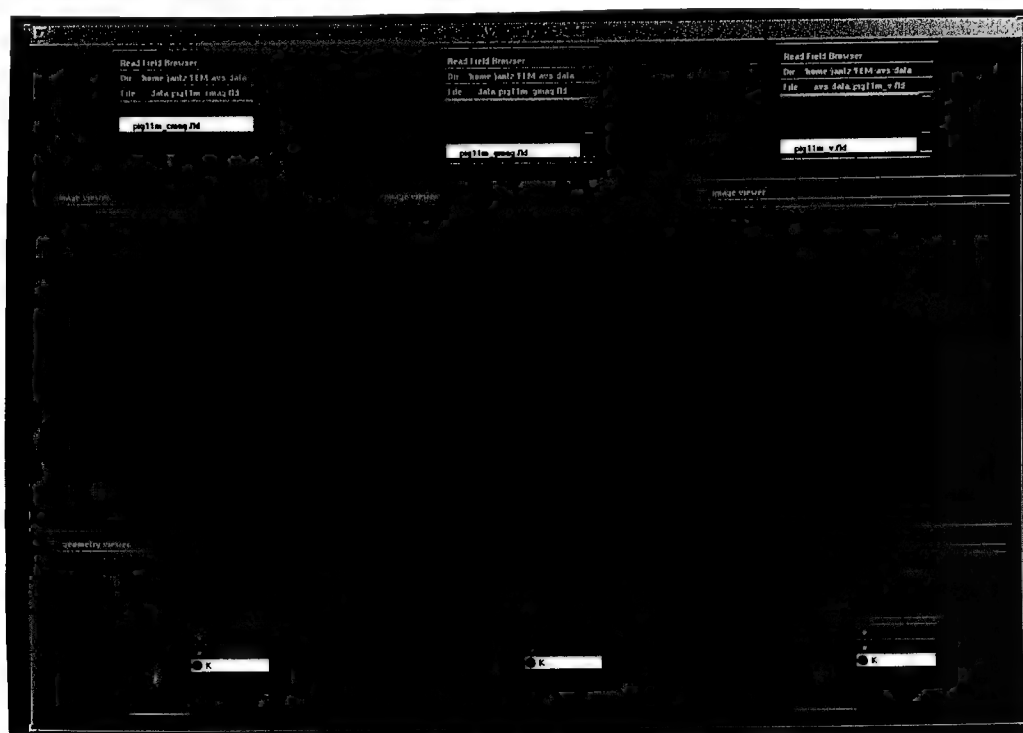
Figure 3.12: Compare fields display - up to three slices
from the same or different data sets are displayed side
by side. The color ranges are scaled according to the
corresponding scalar field.

appropriate slice. The choice of the slice for each field is independent of the other models, so that the same or different slices can be viewed from each model.

3.4 Vector Visualization Tools

The other type of data that we wish to view is vector quantities. In our models, these quantities are the voltage gradient and the current density. AVS modules include three ways of visualizing vector fields which provided good views of our data. These displays will help us investigate the possible correlation between the vector directions and defibrillation efficacy.

3.4.1 Hedgehog arrows

The first vector visualization method is what is known as hedgehog. Its name comes from the looks of the display it generates (Figure 3.13). A plane is positioned in the
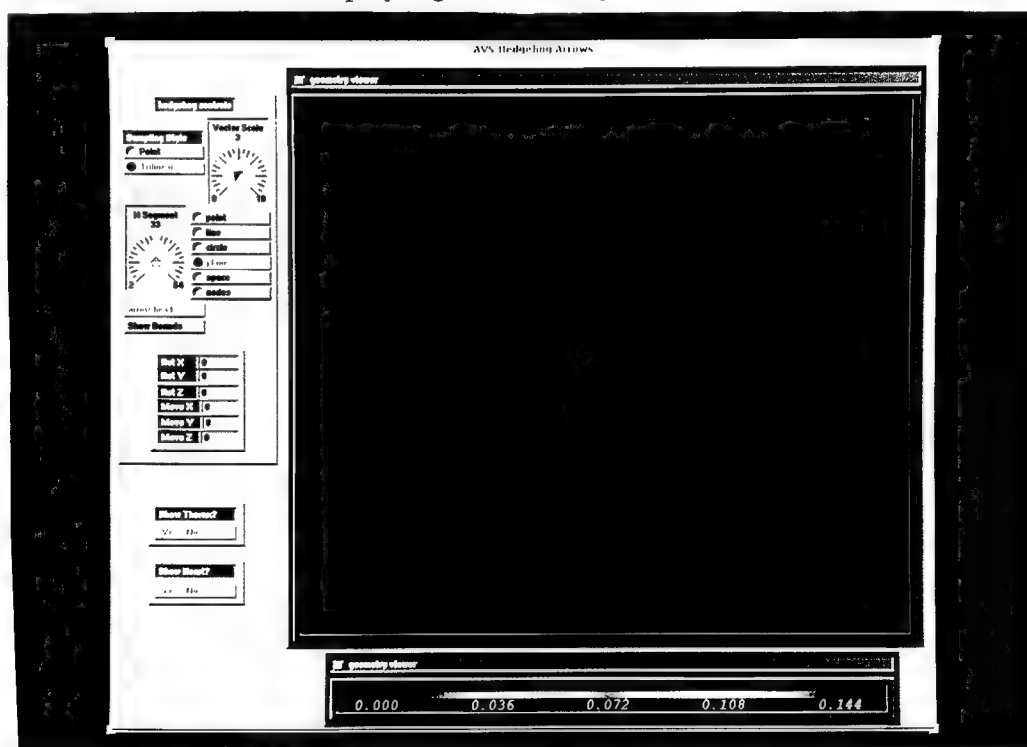
Figure 3.13: Hedgehog arrows display - 3-D arrows are
drawn representing the direction of the vector at a sam-
ple of points in the selected plane. The arrows are dis-
played with identical lengths, and are colored according
to the vector magnitude. A point rendered thorax sur-
face and semitransparent heart surface are included.

model space. From a sample of points in this plane, arrows are drawn showing the direction of the vector field at each point. The arrows are displayed with uniform lengths, while the color of each arrow is determined by the magnitude of the corresponding vector. The user has control over the density of vectors drawn in the plane, and the length of the vectors. The easiest way to move this slice plane in the model is actually a multistage procedure in normal AVS operations (described in Section A.7). Thus, a module was developed to help control the plane. The user can simply input a rotation and/or offset in a given direction, and the plane is moved and the new arrows are calculated. Included in the display is a point-rendered surface of the thorax, and a semitransparent surface of the heart to aid in visualization and analysis. Both of the surfaces can be turned on and off as desired.

3.4.2 Streamlines

The second vector tool is called streamlines. This method also starts with a plane in the model space. Instead of just drawing one arrow at each point, an entire line is drawn which follows the vector field through the model. The overall length of the lines is controllable, as is the step size defining when a new direction for each line is computed. The density of lines drawn from the plane is also user controlled. Each part of the line is colored according to the magnitude of the vector at that given point in space. These static streamlines are displayed in one viewer, along with a point-rendered thorax surface and semitransparent heart surface (Figure 3.14). A second display that animates the formation of the streamlines is included. Short lines are drawn starting from the sampling plane, and then move as each of the steps is calculated along the path of the lines, until the entire line has been traced out. This display also has a semitransparent surface of the heart. All three of the tissue surfaces may be turned on and off. A module similar to the one described in the hedgehog display is included to aid in the maneuvering of the starting plane within the model. Also, since the current density values are usually much lower than the voltage gradient values, a user-defined scaling factor can be applied to the different fields in order to achieve comparable displays.
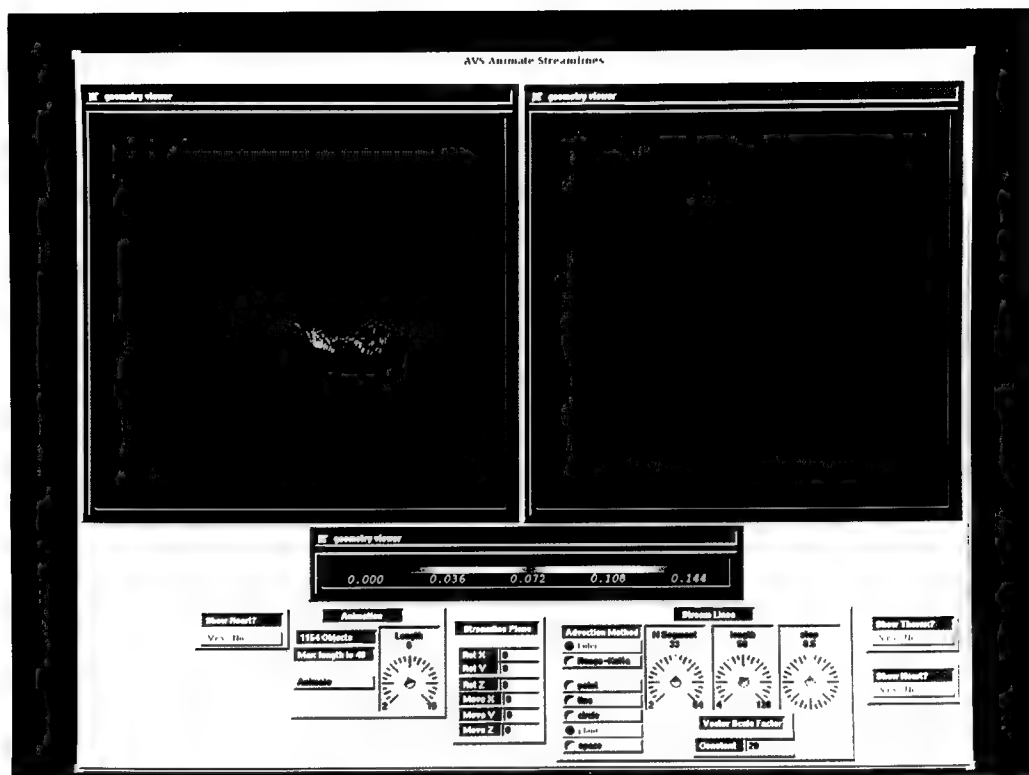
Figure 3.14: Streamlines display - the right display shows a
static set of streamlines drawn from a sampling plane in
the model following the vector field through the thorax.
The left display animates the lines by moving small line
segments along the path of the streamlines. A point-ren-
dered thorax surface is included in the static display, and
a semitransparent heart surface is included in both.

3.4.3 Particle advector

The final vector display method is a vector animation display. Particles are placed
at each point in the starting plane. The particles are then moved through the model as the
vector field dictates. The movement of the particles is continuous until the user stops it.
Once stopped, the particle plane can be moved and a new batch of particles can be
introduced into the model. When the animation is resumed, both sets of particles will
continue moving through the model. When desired, all particles can be erased, and a new
batch injected into the model. The user has control over the size of the particles, and they
can be colored according to the vector magnitudes. There is also a trace function which,
when activated, displays a trail behind each of the particles as they are moved through the

field (Figure 3.15). Again, a point-rendered thorax surface and semitransparent heart surface are displayed, for which the visibility can be controlled. The module giving the user specific control over the position of the plane is included in this network, as is the vector scaling factor control.
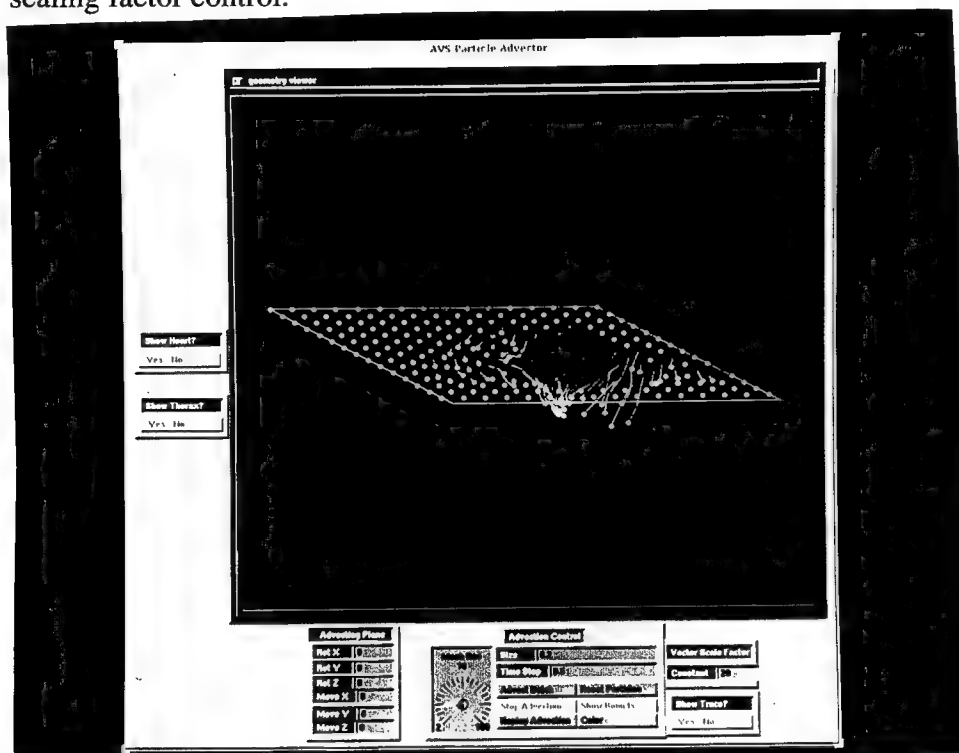
Figure 3.15: Particle advector display - a batch of particles
is released into the thorax and then moved through the
vector field. A point-rendered thorax surface and a
semitransparent heart surface are included. Also shown
are lines tracing the path of each particle.

3.5   Performance

The performance of the visualization tools can best be seen through a simple timing analysis. We compared the speed of the calculations on the IBM RS6000 to those on the SUN SPARCstation 20. The results are in Table 1. For the scalar viewing tools, we timed five different operations: the loading of the network and initialization of the displays with a human thorax model, a one-step 45-degree rotation in the 3-D viewer, the recalculation and display update with a change of parameter (if multiple parameters can be changed, the longest recalculation time is reported), the recalculation and display update

after selecting a new scalar field to view, and the calculation of the value and location of the 3-D probe (if applicable). For the vector viewing tools, the parameter change that takes the longest is the sampling plane manipulation. Thus, two times are reported: a 10-degree rotation, and a 2-slice translation. The 3-D probe is not available in any of the vector tools, so it is not reported. The other three operations timed were the same as in the scalar viewing tools. In order to compare the performance of the SUN and IBM systems, we looked at their respective SPECint92 and SPECfp92 scores. These scores are derived from the results of a set of integer or floating point benchmarks run on each machine. Unfortunately, the IBM RS6000 model 730 that we are using is no longer in production, and thus, was not tested by the SPEC benchmarks. A comparable machine, the IBM RS6000 model 530, scored 28.5 on SPECint and 64.6 on SPECfp. The SUN SPARC-station 20 model 61 scored 93.0 on SPECint and 106.0 on SPECfp. These SPEC results, as well as those for other computer systems, is available in the /pub/spectable file, at ftp.cdf.toronto.edu.

As shown in Table 1, the SUN system outperforms the IBM on most of the operations. In fact, due to a lack of sufficient memory, the IBM is not even able to complete many of the tasks, and thus, no time is reported. As seen in the reported benchmark results, the SUN is capable of higher CPU processing speeds than the IBM. Thus, for compute intensive operations, such as loading a network or computing a tissue surface display, the SUN is superior. On the other hand, the IBM has special hardware available for graphics rendering while the SUN uses all software rendering procedures. This means that the IBM matches or outperforms the SUN for display functions, such as the rotations and probing. Due to these performance results, the SUN SPARCstation 20 was used for the evaluations described in the next chapter, and will continue to be used in the application and future development of our visualization environment.

Table 1: Timing analysis of various operations in the visualization toolset on the SUN SPARCstation and on the IBM RS6000. On the IBM, when no time is recorded, this means that the AVS system was not able to complete the operation due to lack of memory. N/A means not applicable.

Viewing Method on SUN SPARCstation	Time (min:sec) for:				
	loading	rotation	parameter	new file	3D probe
3-slice 2-D	:28	:04	:07	:07	N/A
3-slice 3-D	:30	:06	:07	:09	:08
3-slice 2-D w/ outlines	:56	:04	:17	:25	:03
3-slice 3-D w/ outlines	:54	:06	:14	:26	:06
Arbitrary slicer	1:00	:04	:19	:10	N/A
Colored tissue surface	:43	:07	:22	:22	:13
Heart cut-away	1:03	:01	:49	:25	N/A
Excavated thorax surface	1:01	:10	:28	:20	:13
Excavated volume bounds	:30	:05	:06	:15	:05
Ray trace	:57	:01	:25	:25	N/A
Volume renderer	3:15	7:35	6:01	6:02	N/A
Compare fields	:18	N/A	:00	:06	N/A
	loading	rotation	10° rot.	2 sl. trans.	new file
Hedgehog arrows	1:40	:08	1:37	:26	1:30
Streamlines	3:45	:08	2:15	1:20	3:20
Particle advector	2:00	:07	1:25	:23	1:59
Viewing Method on IBM RS6000	Time (min:sec) for:				
	loading	rotation	parameter	new file	3D probe
3-slice 2-D	-	-	-	-	-
3-slice 3-D	2:47	:03	:28	1:20	:06
3-slice 2-D w/ outlines	2:33	:02	:21	-	:05
3-slice 3-D w/ outlines	3:01	:02	:20	-	:05
Arbitrary slicer	3:28	:03	:30	:37	N/A
Colored tissue surface	2:15	:04	1:20	-	:11
Heart cut-away	-	-	-	-	-
Excavated thorax surface	2:55	:07	2:28	2:40	:15
Excavated volume bounds	1:42	:17	:20	:50	:30
Ray trace	1:59	:05	:38	1:12	N/A
Volume renderer	4:54	6:18	6:31	7:02	N/A
Compare fields	-	-	-	-	-
	loading	rotation	10° rot.	2 sl. trans.	new file
Hedgehog arrows	5:56	:06	-	-	-
Streamlines	-	-	-	-	-
Particle advector	-	-	-	-	-

Chapter 4: Evaluation

In order to evaluate this suite of visualization tools, three approaches have been taken. First, we made an objective evaluation by determining which of the requirements in Section 2.2.2 were fulfilled. Second, a subjective evaluation was made by members of the project group after using these visualization tools. Finally, a comparison has been made to the visualization environments of the University of Utah and Duke University (see Section 2.1).

4.1   Fulfillment of Requirements

It is important to determine how well this project fulfilled the visualization requirements that were identified prior to its beginning. The thirteen requirements of Section 2.2.2 will be treated in the same order, describing if and how each was satisfied. NOTE: Each specific visualization tool will be referred to by its section number from Chapter 3.

1) The off-line data conversion programs are successful in generating the tissue, voltage, current density, and voltage gradient fields from the FEM data. The vector magnitude fields are calculated from the voltage gradient and current density fields by a simple network available on the visualization menu bar. The procedures for performing the data conversions are described in Appendix B

2) Both 2-D and 3-D viewing methods are available. The 3-D methods providing the best views of the overall distributions are the volume rendering techniques of 3.3.10-11. The location of tissues and electrodes is best seen in either 3.3.6 or 3.3.8. Method 3.3.5 is an example of the 2-D investigatory methods, which allows for an arbitrary slice to be extracted from the model. The requirement of combined 2-D and 3-D views is satisfied in 3.3.1-5.

3) The data comparison requirement is fulfilled in 3.3.12. Data from the same or different models can be viewed simultaneously.

4) The ability to extract a subset of tissue types, display the surface around those types, and color the surface according to a scalar value field, is presented in 3.3.6.

5) Extraction of ROIs based on scalar values is demonstrated by the availability of the threshold parameters in the ray-trace method (3.3.10). A mouse-selected ROI capability is not possible with this toolset.

6) A probe is included in many of the viewing methods, both in 2-D and in 3-D. Concurrent movement of the probe in multiple displays was not realized.

7) The tissue-overlay capability is provided in methods 3.3.2 and 3.3.4.

8) Three different vector quantity displays are made available in 3.4.1-3.

9) This set of tools does not currently include the ability to incorporate histograms or other statistical data into the displays.

10) Access to the colormap is made readily available in all of the viewing methods, thus providing the ability to control the range of colors used in the displays. Rotation of objects to change the viewing angle is also possible in all of the 3-D displays. The process of changing the transparency and other properties of an object is possible, but is a multi-step process, as explained in Appendix A (Section A.7). Where possible, point-rendered surfaces were used to avoid the need to change their transparency.

11) Generating hardcopies is possible, but requires a little extra effort. The images must be saved and then read into another network in order to save them in a postscript file. (See Section A.6 in the Appendix)

12) A friendly user interface is included in the visualization environment. The application is menu driven for easy access to the viewing methods. The parameters and controls necessary for each method are provided in the same window as the data viewers. The parameters can be changed with the mouse (a combination of point, click, and drag functions) or with keyboard type-ins. The manipulation of objects in the viewers is possible with combinations of keyboard and mouse controls.

13) On the SUN system, most of the scalar field visualization tools do qualify as being interactive. The rotation and recalculation timing requirements were met in all methods except the volume rendering tool (3.3.11). The volume renderer has taken

over six minutes to recalculate or rotate the model being viewed (a rotation actually includes a recalculation). The vector tools all take longer than 30 seconds for recalculation when the sampling plane is being rotated, or when new fields are chosen, due to an inefficient rotation algorithm and the large amounts of data. See Section 3.5 for a complete timing analysis during various operations in each of the viewing methods.

4.2 Subjective Evaluation and Discussion

Two members of the FEM group were asked to use the visualization suite and provide feedback on the usefulness and completeness of the set of tools. The evaluations were made on the final version of the toolset completed in this project, as a subjective measure of its quality. The suggestions they made, though not immediately implemented, will be addressed in the further development of our visualization environment.

According to the first evaluator, his overall impression was that this was a very useful set of tools. He thought the tools were complete, but could be improved by making them easier to operate. One suggestion in this area was the addition of on-line help menus so that the user does not have to refer to the user's guide when they have questions. Another suggestion followed from a complaint about the rotation speeds. When the mouse is used to rotate an object, a guess must be made as to how far to drag the mouse, because it takes a few seconds to update the display. At times, the evaluator found this confusing and frustrating. He mentioned one way to help would be including a small coordinate axis system in one corner of the display that rotates simultaneously with the dragging of the mouse. When the mouse button is released, the display will then be updated to reflect the rotation. The things that this evaluator liked in the tools were the arbitrary slicer method, the excavation techniques, and the ability to turn the slices on or off in the orthogonal slicer methods (3.3.1-4). One final improvement that he mentioned would be in the methods that include slicing planes. Rather than having to numerically define the slice planes, it would be much more intuitive to be able to use the mouse to drag and position the slice plane in the geometry viewer display.

The other evaluator also had a very good overall impression of the visualization suite. He specifically commented on the usefulness of the tissue outline techniques, the heart cut-away views, and all three of the vector visualization methods. As far as ways to improve some of the methods, one of his suggestions was to develop the ability to use oblique cutting planes in the heart cut-away methods. This way, the heart could be sliced along its own axes rather than the model axes, which is how doctors are used to looking at them. This type of slice is possible in method 3.3.5, so implementing this technique in the heart cut-away method should be the next step. Another improvement that this evaluator suggested was the development of a tool that would make it possible to select, using a menu, a particular tissue to be displayed in the current viewer. For example, in the vector displays, he wanted to be able to view the bones to see how the current was flowing around them. In the same way, a list of displayed tissues should be available from which tissue types can be removed from the display. One more negative aspect he found in the toolset was the required use of floating point numbers when, for our application, integers would work better (such as in the selection of tissue types). Trying to select an integer value on a floating point dial in AVS is very difficult. If an existing control requires a floating point number, and yet we will only want to use integers, this evaluator suggested allowing the input of an integer value which will be automatically converted to a float value for use in the calculations. He agreed with the other evaluator in desiring the overall calculation and display speeds to be improved. One way he suggested would be to limit the size of the model for certain applications. For example, rather than just extracting the surface of the heart from the rest of the model in the heart cut-away method, we could develop a way that only the heart is included in the file read into the network from the beginning. This would significantly decrease the amount of data manipulated by the tools. In general, he was very pleased with the toolset, and said that it would be a great help in the next stages of research.

4.3   Comparison to Other FEM Visualization Environments

As described in Section 2.1, both the research group from the University of Utah, and the group from Duke University designed and programmed their visualization environment on their own. Thus, they were able to design the tools to do exactly what they wanted them to do, in the most efficient way possible. On the other hand, their visualization environment development was much more difficult and took longer than ours did. Due to the use of a commercial package we have encountered some restraints, such as the data file format and the types of program modules available, but the developers of AVS have tried to make sure these restraints are as minimal as possible. Another disadvantage to using AVS is that our tools now rely on software licenses, which cause additional expenses as well as limit the number of users. Our advantage was that we had a lot of visualization capabilities already available for which we did not have to write our own programs. At the same time, when the need arose (and will continue to arise), AVS made it easy to write our own program applications for use in AVS networks.

The key feature of the Utah group's environment is that they divided their needs into two applications. The first set of programs, the "interactive" set, generates lower-quality displays than their other set of programs, but allows the ability to manipulate a geometric object, and the data displayed on it, in real time. What we call our "interactive" tools may be better quality than theirs, but must of ours do not allow real time manipulation. Their second set of programs is not interactive, but generates high-quality, volume-rendered displays. The AVS volume-rendering module does have some of the same potential, but we have not yet developed it fully at this time. The Duke research group's visualization environment emphasizes the high level of control the user has over the mapping of the data to the shading, color, and opacity values in the rendered display. This control also means the learning of a new language, Vexpr, for anyone that wishes to use the tools. For research purposes, this may be acceptable. However, for clinical applications, such as we hope to accomplish, the tools need to be as easy to learn and use as possible. Though our environment may not be as application-specific or computationally-efficient as these others, it does provide us with a high-quality

visualization system that meets our needs, and is flexible and expandable to allow for further development of the tools.

Chapter 5: Conclusion

A suite of tools for the visualization of our FEM models has been developed. Methods for 2-D and 3-D viewing of the scalar and vector quantities have been included. Twelve of the thirteen requirements have been fully or partially met in the visualization environment. The only one that was not included at all was the need for incorporating statistical data into the viewing techniques. Aspects of other requirements, such as the capability of mouse-selecting ROIs and the desire for concurrent probe movement, will need to be developed in further expansion of the visualization tools. The tools were tested by two members of the FEM project group. Their overall impressions were good, and they thought that the toolset would be a big help in future research. One of the disappointments they both mentioned was the slow (non-real-time) speeds achieved by the tools.

Future expansion and improvement of these visualization tools is definitely necessary. As these tools are used, we will begin to determine which are the most helpful and which can be discarded. Control over aspects of the viewing methods which are not currently controllable will be desired. Techniques from different viewing methods may need to be combined into a single method. Also, ideas will be discovered that will lead to the addition of totally new viewing methods. Already some ideas have been proposed. One is the display of not only the scalar values on the surface of a tissue, but a volume averaging through the width of the tissue. Another new idea is the interactive visualization of an approximation of the current path with different surface electrode placements. This will hopefully improve the selection of the initial electrode locations by trying to avoid the ribs which tend to block the current pathway due to their high resistivity values. Significant work must still be done to develop the "pre-solution visualization" techniques (Section 2.3.1), to allow for the introduction of electrodes into the classified tissue models. It has also been proposed that we should investigate the use of parallel computing in order to improve the speed in these visualization tools.

With these improvements and other ideas that are likely to surface, the visualization environment will continue to evolve over the next few years. These tools will help in

the development and evaluation of new solution methods. They will aid in the research of the bioelectric fields generated by new electrode sizes and shapes. They will be valuable in the ongoing investigation to find the specific factors indicating a successful defibrillation. Ideally, the end product will be the incorporation of the final, interactive version of these tools into a clinically useful FEM application to aid in ICD implantation surgery.

References

[1]   J.K. Gilman, S. Jalal, and G.V. Naccarelli, "Predicting and preventing sudden death from cardiac causes," Circulation, vol. 90, pp. 1083-1092, 1994.

[2]   A.S. Manolis, "Transvenous endocardial cardioverter defibrillator systems: is the future here?" (abstract), Archives of Internal Medicine, vol. 154, pp. 617-622 1994.

[3]   G.H. Bardy, B. Hofer, G. Johnson, P.J. Kudenchuk, J.E. Poole, G.L. Dolack, M. Gleva, R. Mitchell, and D. Kelso, "Implantable transvenous cardioverter-defibrillators," Circulation, vol. 87, pp. 1152-1168, 1993.

[4]   C.E. Miller and C.S. Henriquez, "Finite element analysis of bioelectric phenomena," Critical Reviews in Biomedical Engineering, vol. 18, pp. 207-233, 1990.

[5]   D. Blilie Jorgenson, P.H. Schimpf, I. Shen, G. Johnson, D.R. Haynor, G.H. Bardy, and Y. Kim, "Predicting cardiothoracic voltages during high energy shocks: methodology and comparison of experimental to finite element model data," submitted to IEEE Transactions on Biomedical Engineering, 1994.

[6]   N. Shrinidhi, D.R. Haynor, D. Blilie Jorgenson, G.H. Bardy, and Y. Kim, "An efficient tissue classifier for building patient-specific finite element models from x-ray CT images," submitted to IEEE Transactions on Biomedical Engineering, 1994.

[7]   P.H. Schimpf, "A locally adaptive finite element solver for 3D, heterogenous and isotropic boundary value problems in bioengineering," Image Computing Systems Laboratory Internal Report, University of Washington, 1994.

[8]   R.E. Ideker, P.D. Wolf, C. Alferness, W. Krassowska, and W.M. Smith, "Current concepts for selecting the location , size and shape of defibrillation electrodes," PACE, vol. 14, pp. 227-240, 1991.

[9]   L.J. Rosenblum, "Research issues in scientific visualization," IEEE Computer Graphics and Applications, vol. 14, pp. 61-63, March 1994.

[10]   C. Thornborrow, A.J.S. Wilson, and C. Faigle, "Developing modular application builders to exploit MIMD parallel resources," Proc. Visualization '93, pp. 134-141, San Jose, CA, Oct 25-29, 1993.

[11]   S. Bryson, "Virtual reality in scientific visualization," Computers and Graphics, vol. 17, pp. 679-685, 1993.

[12] I.F. Ramirez, S.R. Eisenberg, J.L. Lehr, and F.J. Schoen, "Effects of cardiac configuration, paddle placement and paddle size on defibrillation current distribution: a finite-element model," Medical and Biological Engineering and Computing, vol. 27, pp. 587-594, 1989.

[13] N.G. Sepulveda, J.P. Wikswo, Jr., and D.S. Echt, "Finite element analysis of cardiac defibrillation current distributions," IEEE Transactions on Biomedical Engineering, vol. 37, pp. 354-365, 1990.

[14] A.S.L. Tang, P.D. Wolf, Y. Afework, W.M. Smith, and R.E. Ideker, "Three-dimensional potential gradient fields generated by intracardiac catheter and cutaneous patch electrodes," Circulation, vol. 85, pp. 1857-1864, 1992.

[15] W.J. Karlon, S.R. Eisenberg, and J.L. Lehr, "Effects of paddle placement and size on defibrillation current distribution: a three-dimensional finite element model," IEEE Transactions on Biomedical Engineering, vol. 40, pp.246-255, 1993.

[16] C.R. Johnson, R.S. MacLeod, and M.A. Matheson, "Computational medicine: bioelectric field problems," Computer, vol. 26, pp. 59-67, Oct. 1993.

[17] S. Masse, E. Sevaptsidis, I.D. Parson, and E. Downar, "A three-dimensional display for cardiac activation mapping," PACE, vol. 14, pp. 538-545, 1991.

[18] T.C. Palmer, E.V. Simpson, K.M. Kavanagh, and W.M. Smith, "Visualization of Bioelectric Phenomena," Critical Reviews in Biomedical Engineering, vol. 20, pp. 355-372, 1992.

[19] K.D. Bollacker, E.V. Simpson, G.A. Johnson, G.P. Walcott, K.M. Kavanagh, W.M. Smith, and R.E. Ideker, "A cellular automata three-dimensional model of ventricular cardiac activation," Proc. of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol. 13, pp. 627-628, 1991.

[20] R.S. MacLeod, C.R. Johnson, and M.A. Matheson, "Visualization tools for computational electrocardiology," Proc. Visualization in Biomedical Computing, vol. 1808, pp. 433-444, 1992.

[21] R.S. MacLeod, C.R. Johnson, and M.A. Matheson, "Visualization of cardiac bioelectricity - a case study," Proc. Visualization '92, pp. 411-418, Boston, MA, Oct. 19-23, 1992.

[22] R.S. MacLeod, C.R. Johnson, and M.A. Matheson, "Visualizing bioelectric fields," IEEE Computer Graphics and Applications, vol. 13, pp. 10-12, July 1993.

[23] C. Upson, T. Faulhaber Jr., D. Kamins, D. Laidlaw, D. Schlegel, J. Vroom, R. Gurwitz, and A. van Dam, "The Application Visualization System: a computational environment for scientific visualization," IEEE Computer Graphics and Applications, vol. 9, pp. 30-42, July 1989.

[24] AVS Module Reference, Release 5, Advanced Visual Systems, Inc., Feb. 1993.

[25] AVS Users Guide, Release 4, Advanced Visual Systems, Inc., May 1992.

Appendix A: User's Guide

This user's guide is provided to get you started with the FEM visualization suite, and to explain how to use each of the controls associated with the visualization tools. It assumes that you have already generated the tissue, voltage, voltage gradient, and current density field files for use in AVS. If you have not, refer to Appendix B for instructions on converting the FEM files to AVS fields.

A.1 Getting Started

Since AVS is a licensed software package, the first thing you must do is be sure you are using a computer that is set up with the appropriate license. Then find the "AVS.applns" file. It defines the various applications that are to be made available in the AVS Applications menu. A default file is provided with AVS, but for use with our visualization environment, a copy of this file must specify the "FEM" application network as one of the options and provide the appropriate path to this network. The location of this copy of the AVS.applns file must be defined in the user's ".avsrc" file, along with other AVS environment options, including the appropriate data and network directory paths. Once these files are set up, you can enter the visualization environment.

- type **avs**   (this will start AVS and provide you with a menu of options)
- click on **AVS Applications**   (a new menu will appear)
- click on **FEM**

This gets you into the FEM visualization suite, and brings up the main menu bar at the top of the screen. If the FEM button does not appear on the AVS Applications menu, refer to the AVS Users Guide [25] for more information about setting up your .avsrc and AVS.applns files correctly.

A.2 The Menu Bar

The toolset is operated through the use of a pulldown menu bar. Clicking and holding the left mouse button on one of the menu options drops down a list of choices.

You then drag down to the choice you wish to select, and release the mouse button to make the selection. There are two different menu bar displays. One is the main menu, and the other is a second-level menu.

A.2.1 Main menu

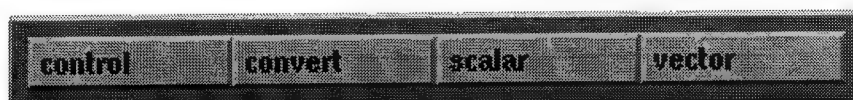The main menu bar (Figure A.1) includes four submenu choices: *control, convert, scalar,* and *vector.*

Figure A.1: Main menu bar

1) *control*: This menu allows you to exit AVS.
2) *convert*: There are five conversion choices in this menu. The first two relate to the fact that the way that the FEM program orients its coordinate axes is different than the way AVS does. It turns out that the y-axis is reversed. Thus, the model must be mirrored in the y-direction to fix the problem. This mirror operation for non-tissue fields is done by the "convert value field" option. For the tissue fields, another conversion is necessary. In order to be consistent between the human and pig models, the tissue types were mapped to a single range of values. This is accomplished in addition to the mirroring operation in the "convert tissue field" network. The third conversion option is to generate the vector magnitude field from a vector field. The final two conversions generate postscript files from previously saved 2-D images or 3-D geometries. See Section A.6 for more information on generating postscript files.
3) *scalar*: This menu includes the 12 choices for viewing the scalar fields (Section 3.3).
4) *vector*: This menu includes the 3 choices for viewing the vector fields (Section 3.4).

A.2.2 Second-level menu

Once a conversion, scalar, or vector tool has been chosen, the menu bar changes (Figure A.2) to include two choices: *control* and *flow.*
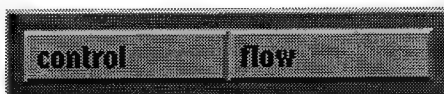
Figure A.2: Second-level menu bar

1) *control*: This menu still allows you to exit AVS. In addition, it allows you to return to the main menu in order to choose another viewing option.

2) *flow*: Though you will not be able to see the underlying processing network, this menu gives you the ability to enable or disable the flow of data through that network. When the flow is on, each time a parameter is changed, the network automatically recalculates the data and updates the display. If you wish to change multiple parameters, you can turn the flow off, and make all the changes at once. Then turn the flow back on to allow recalculation and display with all of the new parameters.

A.3   Displays

Before getting into the actual operation of the tools, we will describe the different data viewers you will be using. There are two types of display windows: the 2-D Image Viewer (or "Display Image" viewer) and the 3-D Geometry Viewer. The various display manipulations can only be carried out on one of each type of viewer window at a time. For each type, the currently selected viewer has its border highlighted red. To select another viewer, click on that border in the desired viewer (thus turning it red).

A.3.1   Image viewer

The image viewer is used to display 2-D images. (The only exception is in the ray-trace display which is viewed in a "Display Image" window. The controls are the same as the "image viewer".) The necessary controls for 2-D image manipulation are *select, translate,* and *scale.*

- *select*: When multiple images are displayed in one image viewer, they are manipulated separately. You can specify which to manipulate by pointing at it and clicking once with the left mouse button. There is no visual cue for which image is selected.

- *translate*: To move the selected image in the image viewer, hold down the right mouse button and drag in the direction you want it moved. When it is positioned correctly, release the mouse button.

- *scale*: To scale up, or zoom in, on the selected image, hold down the SHIFT key and the middle mouse button and drag up or to the right. To zoom out, drag down or to the left.

A.3.2   Geometry viewer

The geometry viewer is used to display 3-D objects. In each of the geometry viewers, the bounds of the model are also displayed. The bounds are color coded to indicate the axis: the red lines are in the x- or i-direction; the green lines are in the y- or j-direction; the blue lines are in the z- or k-direction. The manipulations in a geometry viewer involve *select, translate, scale,* and *rotate.*

- *select*: The geometry viewer opens with the "Top Level" object selected - this includes all of the objects in the view. To select a specific object, point at it and press the left mouse button. Transformations occur only on the selected object. Clicking on the object again returns selection to the Top Level. Very few situations require the selection of a single object. It is recommended that you avoid doing so and leave the Top Level selected.

- *translate*: To move the selected object, hold the right button and drag around the window in the desired direction. Holding the CONTROL key down while dragging limits the translation to one of the major screen axes.

- *scale*: To make the object bigger, hold down the SHIFT key and the middle mouse button and drag up or to the right. To make it smaller, drag down or to the left.

- *rotation*: To rotate the objects, hold down the middle mouse button and drag in the desired rotation direction. Holding down the CONTROL key with the middle mouse button constrains the rotation to one of the major screen axes. Objects can also be rotated with the arrow keys on the keyboard. With the desired geometry viewer selected, the left and right arrows rotate 45 degrees in the horizontal direction, and the up and down arrows rotate 45 degrees in the y-direction.

With any of these manipulations, if the mouse button is released while the mouse is still moving, the object will move continuously until you press any mouse button in the display window.

NOTE: The 2-D views in the "3-slice 2-D with tissue outlines" method (3.3.3) are displayed in geometry viewers. The nature of the contoured slices required this. These views can be zoomed or translated, but rotating them is not necessary, and will only cause confusion.

A.4   File Manipulation

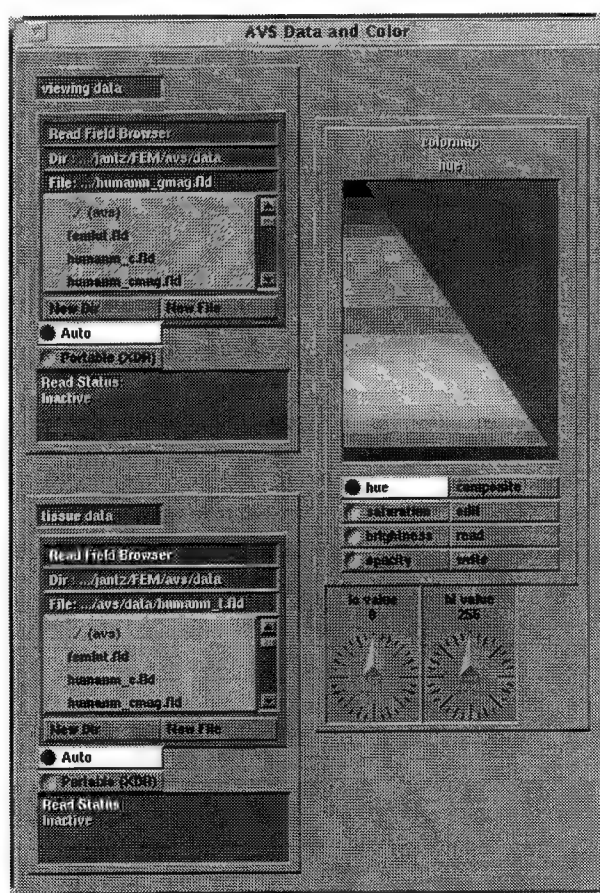Each of the viewing methods starts with the window in Figure A.3 active.

Figure A.3: Data input and colormap window

The colormap controls will be described in Section A.5.1. When a viewing method is chosen, the human thorax model that was used in the development of the toolset is automatically loaded. This gives you an example of what the display looks like, and initializes some of the display environment variables correctly. The first thing you need to do in any method is to choose the data fields you actually wish to view. For each of the data files to be read into the network (tissue data or viewing data), a file browser is presented. This lets you flip through directories and select the field you wish to view. Your data fields must have a ".fld" extension on the filename for them to show up in the file browser. The conversion programs described in Appendix B generate fields with the correct extension. As soon as a data file is selected, it is read into the network. Thus, it is recommended that you turn the network flow off (Section A.2.2), select both the tissue and viewing data files for the new model, and then turn the flow back on.

In the first three conversion networks (see Section A.2.1), a new field is generated and must be written to a file. A file browser is also provided for this. If an existing file is to be overwritten, choose it in the file browser. Otherwise, click on the "new file" button and you are allowed to type in a name for the output field file. Be sure to include the ".fld" extension.

A.5 Controls

Many of the networks in the visualization suite use the same controls, or those which are similar to one another. Thus, in an effort to avoid repetition, this guide does not address each viewing method individually. Instead, it describes, in turn, each type of control that you will find in the entire set of tools. The parameters available in these controls are adjusted using various types of control widgets. These are displayed in Figure A.4. The first type is a dial. The dial can be set by clicking on the desired position on the dial, or by dragging the pointer around the dial. If you point to the middle of the dial, a blue circle becomes visible. Clicking there brings up a control panel that lets you type in a specific value for the parameter. The second type of widget is a slider. This can be thought of as a dial that has been straightened into one line and bounded. The same sort of clicking
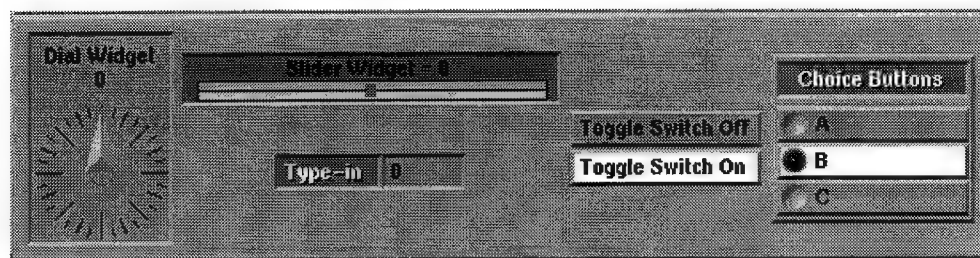
Figure A.4: Parameter control widgets

or dragging can be used to set the parameter value, but no extra control panel is available. A type-in widget allows you to specify the parameter value exactly. The mouse must be pointing at the type-in value window for you to type in the value. A toggle switch is a widget that can be turned on or off by clicking on the button. Lastly, a set of choice buttons may be provided. Only one of the buttons in the set may be active at a given time. These different widgets are used in the following controls, which are found in the visualization tools.

A.5.1 Colormaps

You have control over the colors that are mapped to the values in the chosen viewing data field. The control panel is shown in Figure A.3. The vertical side of the colormap, is an axis ranging from 0 at the top to 255 at the bottom. This range is scaled to the range of the values in the data field. For example, if your data ranges from 0 to 10, the color at a value of 127 in the colormap is assigned to approximately a 5 in your data set. The four choice buttons along the side allow you to choose which aspect of the colormap you wish to display and edit. You can change the hue, saturation, brightness, or opacity values. To edit the color map, you can trace the values directly in the display with the mouse, or you can press the "edit" button and specify exact ranges and values. Each time the colormap is changed, the new colors are automatically calculated for incorporation in the displays. You can also "write" a colormap to a file or "read" in a previously saved one. For more detailed information on colormaps, refer to the "Generate Colormap" module description in the AVS Module Reference manual [24].

A.5.2 Probes

A probe can be attached to an image viewer or to a geometry viewer. For an image viewer, the data display is in a separate window next to the viewer (Figure A.5). To make the probe active, click on the "set pick mode" button of the appropriate probe data display. Then, simply clicking the left mouse button in the viewer will display a crosshair in the viewer at the location of the mouse arrow, and will provide the coordinates and value at that location in the probe window. Holding down the left mouse button while dragging, will continuously update the data display for the moving location of the arrow.
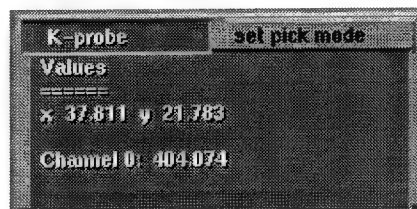
Figure A.5: Image viewer probe data display

In the geometry viewer the data is displayed at the top of the viewer window itself. The probe has a few extra controls (Figure A.6). When the "pick geometry" toggle switch is turned on, a click of the left mouse button tells the probe to go to the location of the object under the arrow. If the toggle is off, the probe is treated like any other object in the viewer, and can be moved or rotated accordingly. The "Sampling Style" menu controls the operation of the probe when a point not on a specific node in the data field is selected. If "point" is chosen, it reports the data at the nearest node. When "trilinear" is selected, an interpolation is done on the data at the eight closest nodes. The "probe type" can be selected as a cursor, a crosshair, or a traditional-looking electronic probe.
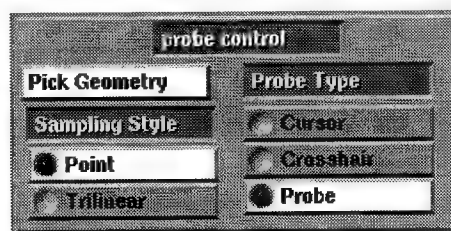
Figure A.6: 3-D probe controls

A.5.3 Orthogonal slicer

This slicer (methods 3.3.1-4,12) takes planes that are orthogonal to one of the dimensional axes of the model (i, j, or k). In our models, the i-slice is a side view from the subject's right side, the j-slice is a front view, and the k-slice is a bottom view. The choice of the number of the slice you wish to view is made by way of a dial widget. In 3.3.12, a set of choice buttons labelled with the three dimensions is included, allowing you the choice of which dimension to slice.

A.5.4 Arbitrary slicer

The arbitrary slicer (3.3.5) has three dial controls. Two of them control the rotation of the slice plane on two of the model's major axes (x and y). The third controls the distance (positive or negative) from the center of the model in a direction orthogonal to the slice plane. These controls can be thought of as defining a location in the polar coordinates, by giving two angles and a distance. The plane is then drawn orthogonal to the polar ray. If all of the parameters are set to zero, it is like viewing the orthogonal slice in the k dimension (from A.5.3) in the center of the model.

A.5.5 Object/Slice visibility

This control is included in some form in most of the viewing methods. It is either a show object "Yes/No" toggle switch or a slice "On/Off" toggle switch (Figure A.7). When the button is turned on, it makes the object or slice visible in the geometry viewer. When the button is turned off, the object is removed from the view.

Figure A.7: Object/Slice visibility controls

A.5.6 Human/Pig selector

In the FEM model, the spacing between pixels in each slice is different than the spacing between the slices. This dimension information is somehow lost in the image viewers (methods 3.3.1,2,4). Thus, a scaling and interpolation function was used to com-

pensate for this error. Choosing the appropriate choice button implements the correct scaling, and displays the correct dimensions of the data.

A.5.7 Update button

The update button is found only in method 3.3.1. When turned on, it connects the parameters to the geometry viewer so that the tissue slices are updated with the 2-D slices. When the button is off, you can quickly scan through the scalar value slices and you do not have to wait for the 3-D calculations to be updated. When the button is turned back on, the tissue slices are snapped to the values in the three slicer parameter control dials.

A.5.8 Tissue outlining

The tissue outlining is accomplished by a contouring algorithm performed on each slice of the tissue field. The three controls for this calculation are shown in Figure A.8.

Figure A.8: Tissue outlining controls

The "contour levels" is a slider that controls how many contour levels are calculated. The "level 0" and "delta" are type-ins that allow you to choose the starting value and step size between the selected number of contour levels. For example, if level 0 was set to 6, delta was set to 1, and contour levels was set to 3, we would get contour lines at 7, 8, and 9. A contour line for a value, m, is drawn between all areas that are less than or equal to m, and all areas that are greater than m. The level 0 value is not included as a contour line.

A.5.9 Tissue selection

This control function (method 3.3.6) simply presents two dials to allow the selection of a minimum and maximum tissue type to be included in the surface generation. For one tissue type, set the minimum and maximum values equal to that tissue type. If both dials are to be changed, it is best to turn the flow off while changing them.

A.5.10 Clipping planes

When this ability is available (methods 3.3.7,10), six dials are presented for you to choose minimum and maximum clipping planes in each of the three dimensions. Behind the scenes, orthogonal slices are positioned at the defined locations, and only the data within these clipping planes is extracted for viewing. The x, y, and z slices correspond to the i, j, and k slices described in Section A.5.3.

A.5.11 Excavate planes

These controls allow you to define the size and location of the rectangular block to be removed in the two excavation techniques (methods 3.3.8,9). Three dials are available to define the cutting planes, thus giving the size of the block. The three "Flip" toggle switches are for defining which corner of the model the block is excavated from.

A.5.12 Ray trace controls

The ray tracer (3.3.10) has many controls of its own which need explanation. The speed of the rendering calculation is largely dependent upon the size of the output image. Thus, the "size" type-ins allow you to change the width and height of the window. Since the ray-trace results in a 2-D image, it cannot be manipulated with the mouse. The rotation and scaling is specified with dial widgets, and a new ray trace is calculated with each new viewing angle. The "threshold values" dial widgets allow you to select only the values in the data field that you want displayed. Everything outside of these thresholds is set to zero. The ray trace also has separate controls for its transparency and perspective. The "alpha scale" determines whether the overall volume is opaque (1.0) or transparent (0.0) or somewhere in-between. (Opacity is also controlled in the colormap - see A.5.1) The "perspective" dial controls whether a perspective (1.0) or non-perspective (0.0) view is displayed.

A.5.13 Rotation control

Since the volume rendering in 3.3.11 is so slow, special dials were added to allow specific rotations to be defined.

A.5.14 Sampling plane controls

Each of the vector visualization methods (3.4.1-3) use a sampling plane to draw the vectors or lines from, or to start the particles from. Using the mouse to move the sampling plane is awkward in normal AVS operations, since a whole new set of calculations must be done for each step in the rotation or translation. Thus, specific type-ins are provided for manipulating the sampling plane (Figure A.9). The units for the rotation parameters is degrees, and for the translation ("move") parameters is number of slices. The manipulations entered occur relative to the current position of the sampling plane. Each type-in must be manually reset to zero if that manipulation is no longer desired. The algorithm used to accomplish these manipulations is inefficient. Be prepared to wait for new calculations (see Table 1 in Section 3.5). Using the "precise transformation" procedure in Section A.6 is not as easy for the user, but is much faster.
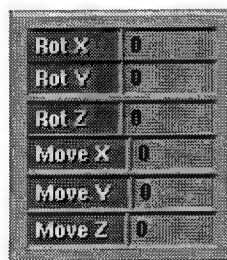

Figure A.9: Sampling plane controls

A.5.15 Hedgehog controls

There are a number of extra controls for the hedgehog viewing method (3.4.1). The "vector scale" dial controls the length of the arrows drawn in the display. There is a choice menu of six possible sampling areas from which arrows can be drawn. The default is a plane. The "N segment" dial controls the density of the arrows drawn. The "Sampling Style" choice determines whether the construction of each arrow is based on the data at the nearest "point," or an interpolation of the eight nearest nodes (trilinear). There are two toggle switches to control whether the "arrow heads" and the "Bounds" of the sampling area are displayed. Finally, the sampling plane controls described in A.5.14 are included.

A.5.16 Streamlines controls

The streamlines viewing method (3.4.2) also includes extra controls. As in A.4.15, you have the choice of the sampling area and the density of the lines drawn. Two more dials are included, allowing you to adjust the overall "length" length of the lines, and the "step" size; i.e. the length each segment is drawn in a given direction before a new vector direction is calculated. Since the magnitudes of the current density vectors are usually so small, the "vector scale factor" is provided as a preprocessing multiplication of the entire vector field. The "advection method" choice determines the method used to calculate the lines. The Euler method is quicker, but the Runge-Kutta technique is more accurate. The sampling plane controls are included, which changes the plane in both the streamlines and the animation. On the animation side, the "length" dial controls the length of the line segments displayed in the animation. Clicking on the "Animate" button begins the animation sequence.

A.5.17 Particle advector controls

A set of controls is included with the particle advector (3.4.3). The "Mesh Res" parameter dial determines the density of particles in each batch. The "size" type-in sets the diameter of each particle. The "Time Step" type-in controls the spacing between successive positions of the particles. "Advect Batch" triggers the release of a batch of particles. When the "Stop Advection" button is turned off, the particles move through the vector space. When that button is on the particles stop but remain in the display until the "Reset Particles" button is pressed. The "Replay Advection" button restarts the advection with the current parameters. The "Color" button allows the particles to be colored according to the magnitude of the vector at its location. The "vector scale factor" and sampling plane orientation controls are available. Finally, each particle leaves a single line "trace" behind it, which can be turned on or off.

A.6 Generating Hardcopies

It is possible to convert either 2-D images or 3-D geometries into postscript files. This capability requires a number of steps to be carried out. The first step is to save the

view in an AVS file. In order to do this, you must enter the appropriate data viewer. At the top left corner of the AVS/FEM visualization environment is the control panel window which includes a menu that looks like this:

Figure A.10: AVS control panel menu

Click and hold the "Data Viewers" button, and a menu will drop down. Select the appropriate viewer depending on whether you want to save an image viewer display or geometry viewer display. This will open up the corresponding data viewer control panel. If you are using the image viewer, find the "write image" button and click on it. You will be prompted for a file name (you must include the entire path and filename), which will be appended with a ".x" extension. If you are using the geometry viewer, find the "save object" button and click on it. Again, you will be prompted for a file name. You will have to press OK twice as a ".geom" file is saved as well as a ".scr" file. Once the file has been saved, return to the main menu by using the visualization "control" menu. In the "convert" menu, choose either "image to postscript" or "geometry to postscript" depending on which display you saved. Once the network is loaded, you will see some "image to postscript controls" and the data viewer. The data viewer control panel should still be visible on the left side of the screen. If you are in the image viewer, choose the "read image" button. If you are in the geometry viewer, choose the "read object" button. You will be given a file browser in which you can locate and select the file you just saved. For the image viewer it should be a ".x" file, and for the geometry viewer it should be the ".scr" file. When the file is selected, the display will be updated with your saved image. Then choose the correct configuration of the postscript file you would like to output and give it a name. Once the name is entered, the postscript file is generated. This postscript file can then be sent to a printer to generate the hardcopy output.

A.7 Advanced Techniques

AVS has many more capabilities that are not included in this user's manual. A couple of these techniques, which may be useful in our FEM application, are quickly presented here:

- *Object Selection*: Using the left mouse button in the geometry viewer to select an object is an arbitrary operation, and there is no easy way of knowing if the object you selected was the one you intended to select. To find out, be sure the correct geometry viewer window is selected, and open the geometry viewer control panel as described in Section A.6. On this panel is a little window that shows a graphical representation of the current object in the geometry viewer display. Clicking any mouse button in the window cycles through the list of objects for the current scene. Above the window is a bar that shows the name of the current object.

- *Transparency Control*: To change the transparency of a given object, you first must make it the current object as described above. Press the "Edit Property" button on the geometry viewer control panel. A window appears that lets you change the display properties of the current object, including its transparency ("Trans").

- *Precise Transformations*: If you wish to rotate or move an object a specified amount, you can do this in the geometry viewer control panel. With the object you want to manipulate selected, press the small dimple on the "Transform Selection" menu bar. This brings up a window of type-ins in which you can specify the exact rotation or translation, either in absolute numbers or relative to the current position of the object. You can also change the "Degree of Rotation" that an object is rotated through when using the arrow keys on the keyboard.

For more information on these and other capabilities please consult the AVS User's Guide [25], especially Chapters 4 and 5 on the Image and Geometry Viewer subsystems.

Appendix B: Conversion Programs

The FEM solution program that we use in our research generates a proprietary type of file format known as a "benny file." It usually has a ".svd" extension. In order to be able to get this data into AVS, we must generate files that are in a format recognized by the AVS software. We have chosen the rectilinear field file format (described in Section 3.1 and in [25]) for our data. The files that we need to generate are the tissue types, voltages, voltage gradients, and current densities. One main problem was encountered in the conversion process. The FEM program reduces the size of each slice from 512x512 to 128x128 in order to make the computation time reasonable for the large amounts of data that we have. In doing so, the tissue conductivities for each 4x4 neighborhood of elements are averaged to get the conductivity for the resulting element. This means that, along the tissue borders, many of the elements will not have a specific tissue class, but instead will be a combination of tissue classes. In generating a field of tissue types, we would then end up with thousands of tissue types rather than the approximately 25 that should exist in the model. In order to solve this problem, the tissue fields must be generated separately from the other fields, starting with the classified CT images. We will first discuss the normal conversions for the three value fields, and then address the more complicated tissue conversion.

A.1 Voltage, Voltage Gradient, and Current Density File Generation

These conversion programs utilize a few of the routines in our FEM solution program environment, so these routines must be available when the conversion programs are compiled. The three field files can be generated all at once, or with three separate commands. The thing to remember is that the files being dealt with can be very large. For example, using a 128x128x114 pig model the file sizes (in bytes) were as follows: original FEM file - 50509340; vector fields - 22415400 each; voltage field - 7473192. The total memory space required is over 100 Mbytes, and the generated files take up almost 50 Mbytes. Thus, if your system does not have enough memory space available, the files

may have to be generated separately and compressed before the next one is created. If you are going to do them all at once, the name of the command is "fem2avs". If you will do them separately, they are called "voltage", "volt_grad", and "curr_dens". The format for running the programs is:

- **\<command\> \<filename.svd\>**

The \<filename.svd\> variable is the name of the FEM file you are getting the data from. The \<command\> is the name of the program you are running. When you run any of these programs, you will be prompted for a file name. Type in a unique and descriptive file name for your data with no extension. Your file name will be appended with a "_v" for the voltage field, "_g" for the voltage gradient field, and "_c" for the current density field. Each file will also be given a ".fld" extension.

A.2 Tissue File Generation

Before the actual process of generating the tissue field, you will need one more file from the conversion programs. If the "fem2avs" file was run, you will notice that it also generated a "_x.fld" file for you. This is a file containing only the coordinates of the elements in the model. If you do not have this file, you must generate one using this command:

- **coords \<filename.svd\> \<avsfile.fld\>**

The \<avsfile.fld\> should be a name that will identify it as a coordinate file (use a "_x.fld" extension).

Once you have this file, as well as all of the classified tissue slices (.cl extension), you are ready to accomplish the following steps to get the tissue file:

1) Use the "cat" command to combine all of the slices, in the correct order, into one big file.

2) Run AVS and enter the "Network Editor." Click on the "read network" button and choose the "512to128a.net" network. You will be presented with a Field Data Interchange Form (Figure B.1). In this form you will have to do a number of things. On the left side of the form a column of choice buttons is presented. First, click on the

Figure B.1: Field Data Interchange Form

"Dimensions" choice button (if it is not already chosen). In the "Value Selection"
window, select dimension 1 and type 512 in the "Enter Value" type-in at the upper
right of the form. Do the same for dimension 2. For dimension 3, enter the number of
slices in your model. Then choose the "Data" choice button. Select the "File Based
Specification" choice button in the upper right corner. Be sure that the "byte" data
type is selected. This completes the inputs on the Data Interchange Form. When this
is done, look at the "file descriptor" control panel on the left of the screen (Figure B.2).
If this is not what you see, find the words "Top Level Stack," and just below them, the
"file descriptor" button, and click on this button. On that control panel, with the
"Browser for File 1" button highlighted, choose the concatenated file that you gener-
ated in step 1 from the file browser. When you press the "send data" button, your file
will be read in and each slice will be downsized into a 128x128 image. Two buttons
below the "file descriptor" control panel button is the "write field" button. After click-
ing on this button, find and press the "new file" button on the control panel. You will
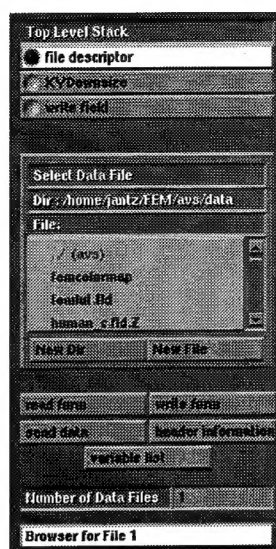be prompted to enter a filename for this new file, and it will then be generated.

Figure B.2: File Descriptor control panel

3) In a separate X-window run a text editor, and strip off the header information of this new file, leaving only the raw binary data.

4) Back in the AVS network editor, press "read network" again, and choose "512to128b.net." Again you will get a Field Data Interchange Form. This time select the "Uniform" choice on the left, and set it to "rectilinear". Choose the "Dimension" button, and set the dimensions again, this time to 128 for dimensions 1 and 2, and the number of slices for dimension 3. Set the "Data" choice information the same way as in step 2, but this time also be sure that "infile 1" is chosen. Next, select the "Points" choice button, and set it to "File Based Specification." Set the data type to "integer" and select the "infile 2" button. In the "Value Selection" window select coordinate 1 and set the "Byte offset" to 0. For coordinate 2, set the "Byte offset" to 512, and for coordinate 3, set it to 1024. In the "file descriptor" control panel, you will have to select two files. With the "Browser for File 1" button highlighted, choose the data field with the stripped header from step 3. With the "Browser for File 2" button highlighted, choose the coordinate field generated by the FEM conversion programs. Press the "send data" button again. This will generate the rectilinear tissue type field of the

same dimensions as the other data fields. Finally, name this new file in the "write field" control panel.

With this done, you will have the four AVS field files necessary to run the FEM visualization suite. See Appendix A for help in the toolset operation.